

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
*Кафедра автоматизованих систем обробки інформації та управління*

«До захисту допущено»  
**В.о. завідувача кафедри**  
О.А.Павлов  
(підпис) (ініціали, прізвище)  
“ ” 2019 р.

**Дипломний проект**  
**на здобуття ступеня бакалавра**

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: «Автоматизована система швидкого реагування на  
екстрені ситуації для мобільної платформи»

**Виконав:** студент 4 курсу, групи ІС-52

Дорогін Ілля Ростиславович  
(прізвище, ім'я, по батькові)

(підпис)

**Керівник**

ст .викл. Солдатова М.О.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з  
графічної  
документації**

ст. викл. Халус О.А.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Рецензент**

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент (-ка) \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) 6.050101

«Комп'ютерні науки» («Інформаційні управляючі системи та технології»)

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

О.А. Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Дорогіну Іллі Ростиславовичу  
(прізвище, ім'я, по батькові)

**1. Тема проекту** « Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи »

керівник проекту Солдатова Марія Олександрівна, ст. викл.  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. № 1181-с

**2. Термін подання студентом проекту** “03” червня 2019 року

**3. Вихідні дані до проекту**

*Технічне завдання*

**4. Зміст пояснювальної записки**

*1. Загальні положення: опис предметного середовища, огляд аналогів, постановка задачі*

*2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних, структура масивів інформації*

*3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування методу розв'язання, опис та приклад виконання алгоритму*

*4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення*

*5. Технологічний розділ: керівництво користувача, методика випробувань*

програмного продукту

## 5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема структурна діяльності користувача

3. Схема бази даних

4. Схема структурна класів програмного забезпечення

5. Схема структурна послідовності здійснення виклику

6. Схема структурна компонентів програмного забезпечення

## 6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «15» лютого 2019 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	16.02.2019	
2.	Аналіз існуючих методів розв'язання задачі	18.02.2019	
3.	Постановка та формалізація задачі	19.02.2019	
4.	Розробка інформаційного забезпечення	20.02.2019	
5.	Алгоритмізація задачі	27.02.2019	
6.	Обґрунтування використовуваних технічних засобів	03.03.2019	
7.	Розробка програмного забезпечення	04.03.2019	
8.	Налагодження програми	10.05.2019	
9.	Виконання графічних документів	15.05.2019	
10.	Оформлення пояснювальної записки	28.05.2019	
11.	Подання ДП на попередній захист	30.05.2019	
12.	Подання ДП на основний захист	03.06.2019	
13.	Подання ДП рецензенту	05.06.2019	

Студент

\_\_\_\_\_ І.Р. Дорогін  
(підпис)

Керівник проекту

\_\_\_\_\_ М.О. Солдатова  
(підпис)

[illegible]



## **Пояснювальна записка до дипломного проекту**

на тему: Автоматизована система швидкого реагування на екстрені ситуації  
для мобільної платформи

---

Київ – 2019 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з шести розділів, містить 113 сторінок, 40 рисунків, 15 таблиць, 1 додатку, 19 джерел.

Дипломний проект присвячений розробці автоматизованої системи швидкого реагування на екстрені ситуації. Цілями розробки є: прискорення процесу виклику екстрених служб, збільшення інформативності даних про виклик, автоматизація процесу диспетчеризації екстрених ситуацій.

У розділі загальних положень детально описано предметне середовище, розглянуто процес діяльності та функціональну модель. Наведено огляд наявних аналогів, а також постановка задачі, де поставлено чіткі цілі розробки та задачі, необхідні для їх досягнення.

Розділ інформаційного забезпечення присвячений опису вхідних та вихідних даних системи, структури бази даних та масивів інформації.

Розділ математичного забезпечення складається з змістовної та математичної постановки задачі про найкоротший шлях, обґрунтування методу розв'язання та опису алгоритму Дейкстри з наведеним прикладом.

У розділі програмного та технічного забезпечення описано засоби розробки, поставлені вимоги до технічного забезпечення, а також детально розглянута архітектура програмного забезпечення.

У технологічному розділі представлено керівництво користувача та описано результати випробування програмного забезпечення.

ЕКСТРЕНИЙ ВИКЛИК, НАДЗВИЧАЙНІ СИТУАЦІЇ,  
ДИСПЕТЧЕРИЗАЦІЯ, НАЙКОРОТШИЙ ШЛЯХ, АЛГОРИТМ ДЕЙКСТРИ

					<b>ДП ІС-5208.1181-с.ПЗ</b>		
		Прізвище	Підпис	Дата			
Розроб.	Дорогін І.Р.				Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи		
Перевірив.	Солдатова М.О.						
Н. кон.	Халус О.А.						
Затв.	Павлов О.А.						
					Літ.	Лист	Листів
						2	
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52		

## ABSTRACT

**Structure and scope of work.** The explanatory note of the graduation project consists of six sections, containing 113 pages, 40 pictures, 15 tables, 1 application, 19 sources.

This graduation project is dedicated to the development of Automated emergency response system for mobile platform.

In the section of the general provisions, the processes of activity, the subject environment and the available analogues of the software product were considered.

In the section of the information support, the input and output data are considered, the database is described.

In mathematical section presents the mathematical formulation of the problem, the general formulation of the problem, algorithms for solving shortest path problem overview, and demonstration of the Dijkstra's algorithm.

In the software section, the structure of classes, their relations and interactions in sequence and the main components of the program are presented. The requirements for the technical support are specified.

The technology section provides user guides and software test methods.

EMERGENCY CALL, EMERGENCY, DISCETTING, SHORTEST PATH,  
DIJKSTRA'S ALGORITHM

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

## ЗМІСТ

ВСТУП .....	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА .....	7
1.1.1 Опис процесу діяльності .....	10
1.1.2 Опис функціональної моделі .....	12
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ .....	14
1.3 ПОСТАНОВКА ЗАДАЧІ .....	16
1.3.1 Призначення розробки .....	16
1.3.2 Цілі та задачі розробки .....	16
Висновок до розділу .....	17
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....	18
2.1 ВХІДНІ ДАНІ .....	18
2.2 ВИХІДНІ ДАНІ .....	19
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....	19
2.4 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ .....	21
Висновок до розділу .....	22
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....	23
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ .....	23
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ .....	23
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ .....	23
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ .....	25
3.4.1 Приклад використання алгоритму .....	26
Висновок до розділу .....	34
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....	35
4.1 ЗАСОБИ РОЗРОБКИ .....	35
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	38

4.2.1	Загальні вимоги .....	38
4.3	АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	38
4.3.1	Діаграма класів .....	41
4.3.2	Діаграма послідовності .....	42
4.3.3	Діаграма компонентів .....	42
4.3.4	Специфікація функцій .....	42
	Висновок до розділу .....	48
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	49
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА .....	49
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	56
5.2.1	Мета випробувань .....	56
5.2.2	Загальні положення .....	56
5.2.3	Результати випробувань .....	56
	Висновок до розділу .....	60
	ЗАГАЛЬНІ ВИСНОВКИ .....	61
	ПЕРЕЛІК ПОСИЛАНЬ .....	63
	ДОДАТОК А.....	75

## ВСТУП

На сьогоднішній день існує велика кількість загроз, які потребують швидкого реагування: від банального отруєння до терористичного акту. Зменшення швидкості реагування на такі ситуації тягне за собою збільшення смертності внаслідок них, а низька інформативність виклику може слугувати чинником неграмотної диспетчиризації, а також може завадити наданню правильної медичної допомоги постраждалим.

Визначено необхідним зменшити час реагування на екстрені випадки на стільки, на скільки це є можливим. В даній роботі пропонуємо автоматизувати функції диспетчера прийому та частково автоматизувати функції диспетчера напрямку.

Дипломний проект присвячений розробці автоматизованої системи швидкого реагування на екстрені ситуації для мобільної платформи. Представлене застосування дасть людям можливість швидко викликати екстрені служби на місце події за допомогою свого смартфона, заздалегідь надавши свої медичні дані.

**Практичне значення одержаних результатів.** Розроблено мобільне застосування для виклику екстрених служб з можливістю задати особисті медичні дані, налаштувати кнопку SOS під особистий сценарій використання та переглядати на карті маршрут бригади екстреного реагування, визначений за допомогою алгоритму Дейкстри.

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Опис предметного середовища

Для реагування на екстрені ситуації в Україні існують такі служби як Пожежна охорона, Поліція, Швидка допомога та Аварійна служба газу. Працюють ці служби за всім відомим принципом: потерпілий звертається за допомогою дзінка на відповідній коротний номер (101, 102, 103, 104) та пояснює ситуацію, що склалася. Але даний спосіб не завжди є оптимальним, а інколи навіть не є можливим, наприклад, для людей з вадами слуху чи голосового апарату. Дана проблема є дуже актуальною в Україні. До сих пір не у всіх областях країни глухонімі люди можуть викликати екстренні служби навіть через смс.

Смертність внаслідок надзвичайних ситуацій в Україні збільшується в більшості випадків через несвоєчасну екстрену допомогу та тривалу комунікацію з аварійними службами. Ефект від скорочення часу на прибуття до місця роботи підрозділів і умовна вартість однієї хвилини в масштабах країни спеціалістами обчислюється в мільйони євро на рік[1].

Для прикладу розглянемо роботу диспетчерської служби екстреної медичної допомоги (всі екстрені служби працюють за аналогічним принципом).

Оперативно-диспетчерська служба центру забезпечує екстрену медичну диспетчеризацію шляхом систематизованих заходів з прийому, обробки викликів та визначення пріоритетів для своєчасного відправлення на місце події необхідних сил та засобів[2].

Об'єкти диспетчеризації представлені на рисунку 1.1.

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7



Рисунок 1.1 – Об’єкти диспетчеризації

Для отримання інформації про виклики у складі диспетчерської служби центру визначаються диспетчери з приймання викликів, які здійснюють:

- приймання виклику;
- заповнення первинної медичної документації в електронному вигляді;
- передавання інформації в електронному вигляді диспетчеру напрямку.

Диспетчери напрямків забезпечують:

- приймання інформації (електронної картки) від диспетчера з приймання викликів;
- передавання інформації (електронної або паперової картки) керівникові бригади;
- отримання від бригади підтвердження щодо надходження інформації про виклик;

Змн.	Арк.	№ докум.	Підпис	Дата



- інформаційний супровід на всіх етапах надання екстреної медичної допомоги за викликом;
- передання до стаціонарних закладів охорони здоров'я інформації щодо необхідності госпіталізації пацієнтів за викликами;
- отримання повідомлення про обсяг екстреної медичної допомоги, наданої за викликом;
- передання інформації про завершення бригадою надання зазначеної допомоги за викликом до диспетчерської служби центру;
- залучення додаткових бригад до надання екстреної медичної допомоги за наявності великої кількості постраждалих на місці події[3].

На основі вищевказаного представимо функції диспетчерів на рисунку 1.2.

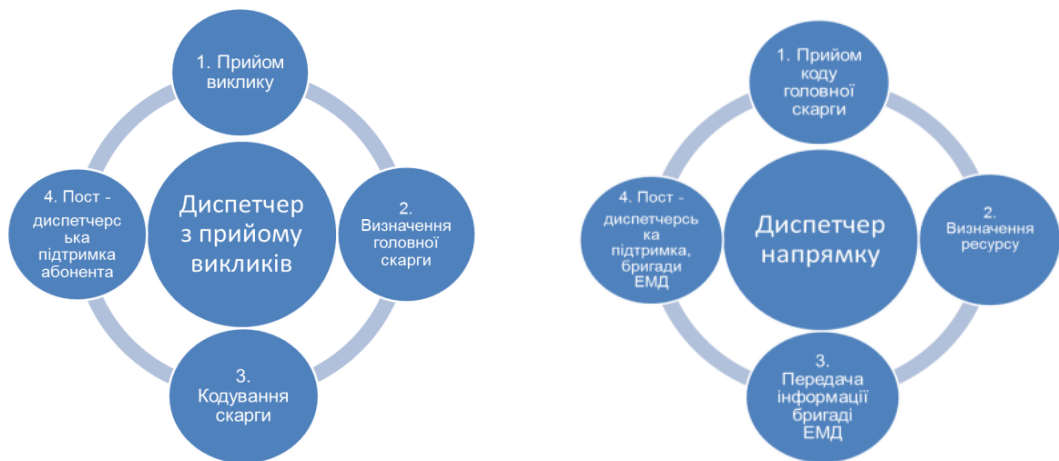


Рисунок 1.2 – Функції диспетчерів прийому та напрямку

За допомогою розробленого у представленій роботі застосування зможемо повністю автоматизувати роботу диспетчеру прийому викликів, а також частково полегшити роботу диспетчера напрямку. В майбутньому при розробці повноцінної системи реагування сумісно з держаними органами можливо звести роботу диспетчера напрямку до мінімуму. Створення такої системи зараз є найбільш актуальним, тому що саме зараз йдуть роботи по впровадженню єдиної служби порятунку 112. Система 112 дозволить інтегрувати в собі до 40 державних і муніципальних підрозділів екстреної допомоги населенню, а створення єдиного диспетчерського центру

					ДП ІС-5208.1181-с.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

полегшить інтеграцію такої системи у вже існуючу, а також збільшиться кількість об'єктів, що можуть бути автоматизовані.

Суть даної роботи полягає в створенні застосування призначеного для користувачів (абонентів), яке має пришвидшити та якісно покращити взаємодію людей з екстреними службами. Екстреним службам допоможе швидше реагувати на екстренні випадки та більш детально формувати знання про конкретну ситуацію. Користувачі, в свою чергу, зможуть зручно та швидко викликати потрібну службу, будучи впевненими, що спеціалістам надана актуальна та вичерпна інформація, а також отримують інформацію про надання першої допомоги та дії при різних надзвичайних ситуаціях.

За допомогою розробленого застосування можна буде викликати екстренні служби на актуальне місце подій, яке визначатиметься автоматично службами геолокації, задати особисті медичні дані, які можуть бути потрібними екстреним службам для надання правильної допомоги (наприклад, наявність алергічних реакцій на ті чи інші речовини), а також отримати інформацію про надання першої допомоги та дії при різних надзвичайних ситуаціях. Якщо потрібно викликати допомогу для іншої людини (якщо є потреба надати дані потерпілої людини), або групи людей, можна скористатися одним з доступних варіантів: ввести ID потерпілого, або ввести номер потерпілого (чи обрати зі списку контактів).

### 1.1.1 Опис процесу діяльності

Розглянемо процес діяльності системи, його описано за допомогою діаграми діяльності.

Структурна схема діяльності користувача наведена в графічному матеріалі.

На першому етапі відбувається реєстрація, користувачі повинні ввести свої особисті дані.

					ДП ІС-5208.1181-с.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Вхідна інформація на цьому етапі:

- прізвище;
- ім'я;
- по-батькові;
- дата народження;
- місто;
- місце проживання;
- номер телефону;
- електронна пошта;
- пароль

Якщо реєстрація не відбулась, то слід перевірити дані.

В майбутньому планується реалізація реєстрації за допомогою BankID, що дасть можливість легко ідентифікувати особистість користувача. Для організацій, що надають адміністративні послуги населенню, сервіс надається безкоштовно[4].

Авторизація потребує таких вхідних даних:

- електронна пошта;
- пароль.

Увійшовши у систему, користувач може переглядати довідкову інформацію, змінювати особисту інформацію та викликати обрану екстренну службу. Під час виклику користувач задає тип ситуації, може надати інформацію у текстовому вигляді та додати медіафайли. Також користувач має можливість додати постраждалого (за замовчуванням обраний лише той користувач, з акаунту якого відбувається виклик). Після виклику користувач потрапляє на сторінку очукування, де йому пропонується довідкова інформація по його типу проблеми, а також доступний чат із диспетчером для обміну інформацією по даному виклику.

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

### 1.1.2 Опис функціональної моделі

На даному етапі розробки системи функціонує єдиний актор — користувач. Можливі дії користувача представлені на схемі варіантів використання.

Структурна схема варіантів використання наведена в графічному матеріалі.

Основні варіанти використання користувача детально описані у таблиці 1.1.

Таблиця 1.1– Опис дій варіантів використання

Актор	Варіант використання	Опис дії варіанта використання
Користувач	Перегляд довідникової інформації	Користувач має змогу переглянути інформацію про надання першої допомоги та дії при різних надзвичайних ситуаціях
	Вибір екстренної служби	На головному екрані користувач обирає потрібну екстренну службу (Пожежна охорона, Поліція, Швидка допомога та Аварійна служба газу).
	Вибір типу ситуації	В залежності від екстренної служби користувач має змогу обрати певний тип ситуації
	Надання інформації	Користувач має змогу додати додаткові матеріали про ситуацію у вигляді тексту, фото, відео.
	Вибір постраждалих	Якщо відомо, кому саме потрібна медична допомога, користувач має змогу додати людей до списку постраждалих: обрати зі списку контактів, ввести номер мобільного людини, або ввести ID людини.

## Продовження таблиці 1.1

Актор	Варіант використання	Опис дії варіанта використання
	Комунікація з диспетчером	Користувач має змогу обмінюватись повідомленнями в чаті з диспетчером, а там же переглядати довідкову інформацію по обраній ситуації.
	Виведення ID на екран блокування	Користувач має змогу вивести особистий ID на екран блокування телефону для полегшення дій в разі надзвичайних ситуацій.
	Введення особистих медичних даних	Користувач має змогу вказати особисті медичні дані для використання їх швидкою допомогою у випадку, якщо користувачу буде потрібна допомога.
	SOS виклик	В разі екстренної ситуації користувач має змогу просто натиснути кнопку SOS та будуть виконані заздалегідь задані дії.
	Настройка кнопки SOS	Користувач має змогу настроїти дії, що будуть виконані після натискання кнопки SOS (якщо людина має захворювання на кшталт епілепсії вона заздалегідь знає найбільш ймовірний випадок натискання кнопки): <ul style="list-style-type: none"> <li>- задати повідомлення, що буде відправлене;</li> <li>- задати номери та служби, на які буде відправлене повідомлення.</li> </ul>

## 1.2 Огляд наявних аналогів

В пердсталеного застосування небагато аналогів, більшість з яких мають обмежений функціонал, а що саме головне — вони не працюють в Україні. Розглянемо їх та виділимо основні переваги та недоліви у поривнянні з нашим застосуванням.

### Прямий зв'язок з поліцією Дубая через m-Govt

Через m-Govt зареєстровані користувачі можуть безпосередньо звернутися до поліції Дубая. За допомогою застосування користувачі можуть дізнатися інформацію про оплату штрафів або відправити фотографії на підтвердження своїх скарг на неправомірні дії поліцейських. За допомогою телефону в найближчому часі можна буде оформити заявку на видачу довідки про несудимість. Зараз ця послуга доступна тільки громадянам ОАЕ.

Громадяни на яких заведено справу зможуть через свій смартфон відправити запит в поліцію і отримати інформацію про хід його розгляду. Ще в цьому додатку для смартфона передбачений екстрений виклик, що забезпечує прямий канал зв'язку з поліцією. В екстреному випадку користувачеві досить натиснути кнопку SOS або активізувати сеанс зв'язку за допомогою спеціальної голосової команди. Як голосовою команду, за умовчанням, використовують фразу «Допоможіть!» англійською або арабською мовою. Оператор колл-центру поліції Дубая при отриманні екстреного дзвінка негайно передзвонить людині, яка потребує допомоги. Щоб користуватися функціями цього застосування обов'язково потрібно пройти процедуру реєстрації. Для реєстрації необхідно мати при собі посвідчення особи, оскільки всі дані користувача системи будуть зв'язатися з базою даних Emirates ID Authority. Туристам також надається можливість частково скористатися цією системою для зв'язку з службою забезпечення безпеки туристів[5].

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

## Переваги:

- можливість взаємодії з поліцією в різних питаннях, а не лише виклик екстренної допомоги

## Недоліки:

- не має можливості вказати особисті медичні дані

**Help Me у Дубаї**

Ідея застосування базується на участі в надзвичайних ситуаціях волонтерів, які мають право надавати медичну допомогу пацієнтам. Волонтери можуть прибувати на допомогу за допомогою карти, яка повідомить місце розташування пацієнта і його симптоми. Метою є скорочення часу реагування на надзвичайні ситуації в Дубаї, здатне врятувати життя людей, які страждають хронічними захворюваннями, і пацієнтів з серцевими нападами і інсультами[6].

## Переваги:

- можливість реагування волонтерів, які можуть прибути на місце подій набагато швидше екстрених служб

## Недоліки:

- не має можливості виклику саме екстрених служб

Цей додаток не є прямим аналогом представленого застосування, але є найшвидшою альтернативою швидкої допомоги. Дана ідея може стати основою подальших розробок мого застосування.

**112 МО**

За допомогою мобільного додатку «Система-112 МО» жителі Підмосков'я можуть повідомити про надзвичайні ситуації і викликати необхідні екстрені служби. Додаток має широкий функціонал. Натиснувши на кнопку в центрі екрану, можна зателефонувати оператору системи 112.

Крім оперативного виклику екстрених служб через мобільний додаток системи 112 користувачі можуть розміщувати повідомлення про подію на інтерактивній карті. Також на інтерактивній карті передбачений пошук

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

найближчих екстрених служб і держустанов. Наприклад, можна знайти адресу найближчої лікарні, травмпункту, ЄДДС, МФЦ.

Також в мобільному додатку розміщена довідкова інформація про способи надання першої допомоги і діях при різних надзвичайних ситуаціях.

Алгоритм роботи системи 112 дозволяє оператору, який приймає дзвінок, одночасно викликати на місце події і координувати роботу кількох екстрених служб[7].

Переваги:

- повністю працююча на рівні держави система

Недоліки:

- діє лише на території Підмосков'я.

### 1.3 Постановка задачі

#### 1.3.1 Призначення розробки

Призначенням розробки є створення системи швидкого реагування на екстренні ситуації.

#### 1.3.2 Цілі та задачі розробки

Цілями розробки системи є:

- прискорення процесу виклику екстрених служб;
- збільшення інформативності даних про виклик;
- автоматизація процесу диспетчеризації екстрених ситуацій.

Для досягнення поставлених цілей необхідно реалізувати наступні задачі.

- виклик екстрених служб
- пошук найближчого закладу потрібного типу та побудова найкоротшого маршруту для бригади екстреного реагування
- перегляд маршруту бригади на карті

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16



- чат для комунікації з диспетчером;
- перегляд довідкової інформації;
- можливість створення власної медичної картки;
- кнопка SOS з можливістю її настройки.

Реалізація поставлених задач створить оптимально-зручне середовище для виклику екстрених служб через мобільне застосування.

### **Висновок до розділу**

У даному розділі виконаний детальний аналіз предметного середовища на основі дослідження роботи диспетчерів прийому викликів та диспетчерів напрямку, задіяних в реагуванні на екстрені ситуації, а також побудованих діаграм діяльності та варіантів використання.

Проаналізовано наявні аналоги системи, у порівнянні з якими виділені основні переваги та недоліки розробленого застосування.

Поставлено цілі та задачі до виконання для розробки автоматизованої системи швидкого реагування на екстрені ситуації.

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Вхідні дані представлені у таблиці 2.1.

Таблиця 2.1 – Опис вхідних даних.

Данні	Опис
Особисті данні	Дані користувачів, які заповнюються у формах реєстрації (прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль) та авторизації (електронна пошта, пароль). Слугують для створення та входу у систему користувачів
Особисті медичні дані	Користувач створює свою медичну картку та має можливість заповнити такі дані: хвороби, алергічні реакції, медикаменти, група крові, вага, зріст Ці дані будуть передані медичному закладу, що реагуватиме на виклик.
Інформація про виклик	Користувачі заповняють форму з інформацією по ситуації, що склалася: тип ситуації, додаткова інформація (у вигляді тексту, аудіозапису, фото, відео), постраждалі.

## 2.2 Вихідні дані

Вихідні дані представлені в таблиці 2.2.

Таблиця 2.2 – Опис вихідних даних

Дані	Опис
Довідкові дані	Після виконання користувачем виклику, йому автоматично пропонується довідкова інформація актуальна для його ситуації.
Дані про виклик	Уся інформація про виклик, що буде передана диспетчеру, а саме: код ситуації, надана користувачем інформація про виклик, медичні картки постраждалих, найближчий заклад потрібного типу, оптимальний маршрут для бригади екстреного реагування.

## 2.3 Опис структури бази даних

В створеній базі даних сформовані таблиці: «User», «MedicalRecord», «Result», «Photo», «Injured», «SituationType».

Опишемо призначення кожної із них:

- User — таблиця, яка містить інформацію про користувача;
- MedicalRecord — таблиця, яка містить інформацію про медичну картку користувача;
- Result — таблиця, яка містить інформацію про зроблений виклик
- Photo — таблиця, яка містить фотографії, додані до виклику
- Injured — таблиця, яка містить інформацію про постраждалих,

доданих до виклику

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

– SituationType — таблиця, яка містить інформацію про типи ситуацій  
Схема бази даних наведена в графічному матеріалі.

Таблиця 2.3 – Детальний опис таблиць БД

Назва таблиці	Назва стовпця	Тип даних	Опис поля
User	ID	integer	Ключ
	Name	string	Ім'я
	Surname	string	Прізвище
	Birthday	date	Дата народження
	Address	string	Домашня адреса
	Phone	string	Телефонний номер
	Email	string	Електронна пошта
MedicalRecord	ID	integer	Ключ
	UserID	integer	Посилання користувача
	Diseases	string	Хвороби
	Allergic_reactions	string	Алергічні реакції
	Medicines	string	Медикаменти
	Blood_type	string	Група крові
	Growth	integer	Зріст
	Weight	integer	Вага
Result	ID	integer	Ключ
	SituationTypeID	integer	Посилання на тип ситуації
	Info	string	Інформація надана користувачем
	Location	string	Координати місця події
	Route	string	Маршрут бригади до місця події
	FireDepartment	boolean	Задіяти Пожежну службу

## Продовження таблиці 2.3

Назва таблиці	Назва стовпця	Тип даних	Опис поля
	Police	boolean	Задіяти Поліцію
	Ambulance	boolean	Задіяти Швидку допомогу
	GasEmergency	boolean	Задіяти Екстрену службу газу
Photo	ID	integer	Ключ
	ResultID	integer	Посилання на результат виклику
	PhotoBM	blob	Фотографія представлена у вигляді байтового масиву
Injured	ID	integer	Ключ
	ResultID	integer	Посилання на результат
	UserID	integer	Посилання на користувача
SituationType	ID	integer	Ключ
	Name	string	Назва типу ситуації

## 2.4 Структура масивів інформації

Дорожний граф міста, в якому дорога виступає ребром, а перехрестя — вершиною, зберігається у вигляді файлу graph.json, який має наступну структуру:

```
{
  "region" : "Kyiv",
  "nodes" : [ { "id" : "1", "x" : "1.111111", "y" : "1.111111" }, ... ],
  "edges" : [ { "fromId" : "1", "toId" : "2", "weight" : "2.12" }, ... ],
}
```

**Висновок до розділу**

У даному розділі описано, які вхідні дані поступають від користувача системою швидкого реагування на екстрені ситуації та які дані віддає система на виході.

Була побудована схема структури бази даних, використана для збереження даних системи. Описано кожну таблицю та поля таблиць.

					ДП ІС-5208.1181-с.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Після виклику екстренної служби користувачем, з'являється задача пошуку найближчого закладу потрібного типу та найкоротшого шляху до нього.

Маємо план міста з нанесеними на нього місцями розміщення пожежних частин, лікарень тощо. Знайти найближчий до поточного місцезнаходження заклад потрібного типу.

#### 3.2 Математична постановка задачі

Дана сітка автомобільних доріг:

- орієнтований зв'язний граф  $G=(V,E)$ , де
  - $V$  — множина перехресть (вершин),
  - $E$  — множина доріг (ребер);
- $w_{ij} \geq 0$  — довжина дороги (вага ребра),
- $|V|=n, |E|=m$ ,
- початкова точка — вершина  $s$ , пункти призначення —  $V_k \subset V$  (множина кінцевих вершин)

Ціль: знайти найкоротший шлях до найближчої вершини з  $V_k$ .

#### 3.3 Обґрунтування методу розв'язання

Задача про найкоротший шлях — це завдання пошуку такого маршруту між двома вершинами графа, сума ваг ребер, які входять до маршруту, найменша[8].

В теорії графів задача про найкоротший шлях вважається однією з найважливіших серед класичних задач. На практиці повсюдно використовується у геймдеві, мережах та під час навігації, як і в

представленій роботі. В цьому випадку перехрестя виступають в ролі вершин, а дороги — в ролі ребер.

На сьогоднішній день існує багато алгоритмів, що розв'язують дану задачу. З тих, що

- алгоритм Дейкстри — знаходить найкоротший шлях від однієї вершини до всіх інших вершин графа, ребра якого не мають від'ємної ваги;
- алгоритм Белмана-Форда — знаходить найкоротший шлях від однієї вершини до всіх інших вершин графа, який може мати від'ємні ваги
- $A^*$  — знаходить найдешевший маршрут між двома точками по алгоритму першого найкращого варіанту
- алгоритм Флойда-Уоршела — знаходить найкоротші шляхи між усіма вершинами графа
- алгоритм Джонса — знаходить найкоротші шляхи між усіма вершинами графа

Так як від'ємних ваг ребер в нашому випадку не буде, а шукати маршрут ми будемо з однієї початкової точки, то алгоритми Белмана-Форда, Флойда-Уоршела та Джонса нам не підходять. Для алгоритму  $A^*$ , в свою чергу, потрібно задати евристичну функцію оцінки відстані до шуканої вершини. А в нашому випадку, коли ми шукаємо шлях не до однієї, а до декількох вершин, які можуть бути розташовані в безлічі напрямків, таку функцію задати неможливо. Тобто для нашої системи доречним буде вибір алгоритму Дейкстри.

Алгоритм Дейкстри — алгоритм пошуку найкоротшого шляху з однієї вершини до всіх інших вершин графа без ребер від'ємної ваги, відкритий Дейкстрою.

Складність алгоритму Дейкстри складається з двох основних операцій: час знаходження вершини з найменшою величиною відстані  $d[v]$ , і час здійснення релаксації, тобто час зміни величини  $d[to]$ .



Для розрахунку складності алгоритму Дейкстри, розглядаємо дві операції: час знаходження найближчої вершини -  $O(n)$ , та час релаксації -  $O(1)$ . Так як перша операція виконується  $O(n)$  раз, а друга -  $O(m)$ , асимптотика алгоритму Дейкстри при простій реалізації —  $O(n^2 + m)$ . Таким чином при збільшенні  $n$  (збільшенні розрідженості графа), отримана оцінка стає менш оптимальною. Тобто, для зменшення часу виконання алгоритму потрібно намагатись зменшити час виконання операцій першого типу. Задля цього зазвичай використовують такі структури даних як фібоначчійові купи, які дозволяють скоротити час цих операцій до  $O(\log n)$ .

Тобто при використанні алгоритму Дейкстри маємо можливість в майбутньому модифікувати алгоритм та приблизитись до теоретичного мінімуму часу алгоритму пошуку найкоротшого шляху —  $(n \log n + m)$ .

### 3.4 Опис методів розв'язання

Дано орієнтований граф  $G=(V,E)$ , де  $V$  — множина вершин та  $E$  — множина ребер, з початковою вершиною обходу  $v_0$ , кінцевими вершинами  $V_k \in V$  та вагами  $w_{ij}$  для кожного ребра  $(i, j)$  в  $E$ . Нехай,  $d_i$  — відстань від вершини  $v_0$  до вершини  $v_i$ ,  $D_i$  — потенційна відстань від  $v_0$  до вершини  $v_i$ ,  $U$  — множина вершин до яких відстань відома,  $v_c$  — поточна вершина.

За постановкою задачі нам потрібно знайти найближчу з кінцевих вершин та відстань до неї.

Алгоритм Дейкстры

1.  $d_0 = 0$ .  $D_i = \infty$  для всіх  $v_i \in V \setminus U$
2.  $U = \emptyset$
3. Оголошуємо вершину  $v_0$  поточною ( $c$ ).
4. Для всіх  $v_i \in V \setminus U$  суміжних з  $c$  перераховуємо потенційні відстані  $D_i = \min\{D_i, d_c + w_{ci}\}$ .
5. Якщо  $\exists v_i \in V \setminus U$ , для якої  $D_i = \min_{v \in V \setminus U} D$  ( $\exists v_i \in V \setminus U$  для якого  $D_i \neq \infty$ ), то оголошуємо  $v_i$  поточною, а відстань до неї стає остаточною  $d_i = D_i$ ,

$U = U \cup \{v_i\}$ . Інакше, маршруту не існує, алгоритм завершено.

6. Якщо  $v_c \in V_k$ , то  $v_c$  — шукана вершина, а відстань до неї дорівнює  $d_c$ .

Алгоритм завершено.

7. Перейти до кроку 4.

### 3.4.1 Приклад використання алгоритму

Дано неорієнтований зважений граф  $G(V, E)$ , зображений на рисунку 3.1. Маємо початкову точку у вершині  $v_3$ , та три кінцеві точки у вершинах  $v_7$ ,  $v_{12}$  та  $v_{19}$ . Потрібно знайти відстань до найближчої з кінцевих точок.

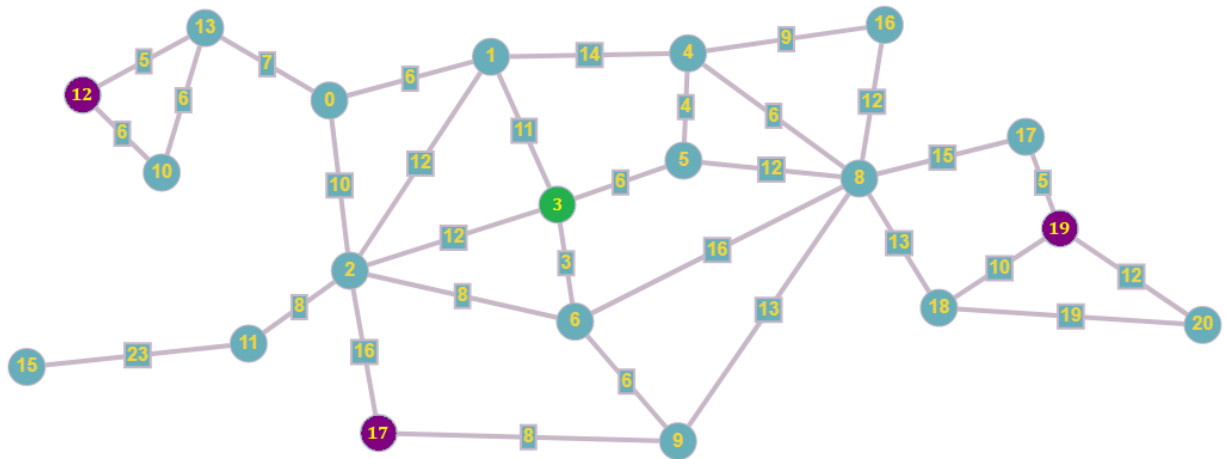


Рисунок 3.1 – Граф  $G(V, E)$

#### Ініціалізація

На початку роботи алгоритми, відстані до всіх вершин графа прирівнюємо до нескінченності, а до початкової — 0. Всі вершини помічються неопрацьованими — остаточні відстані до них ще не відомі.

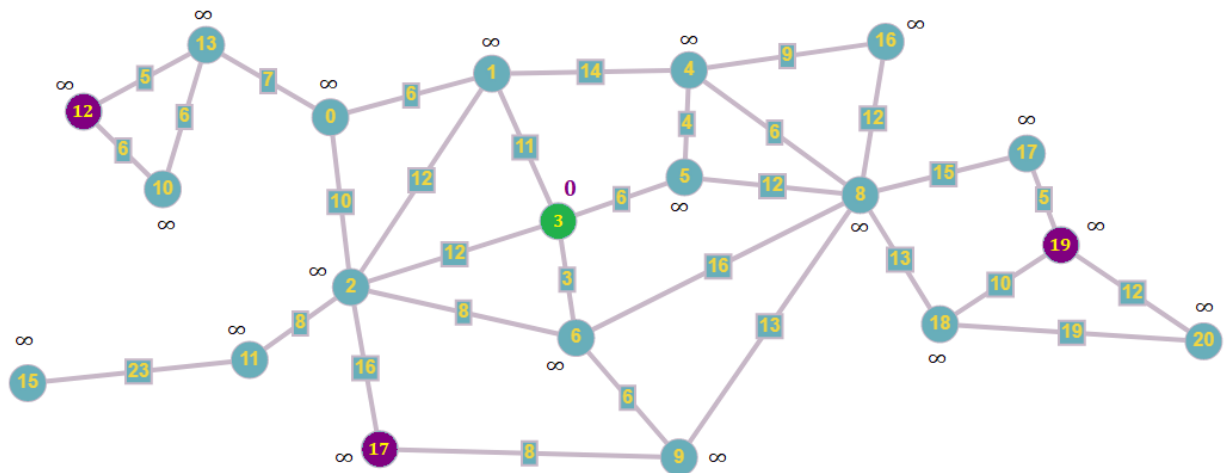
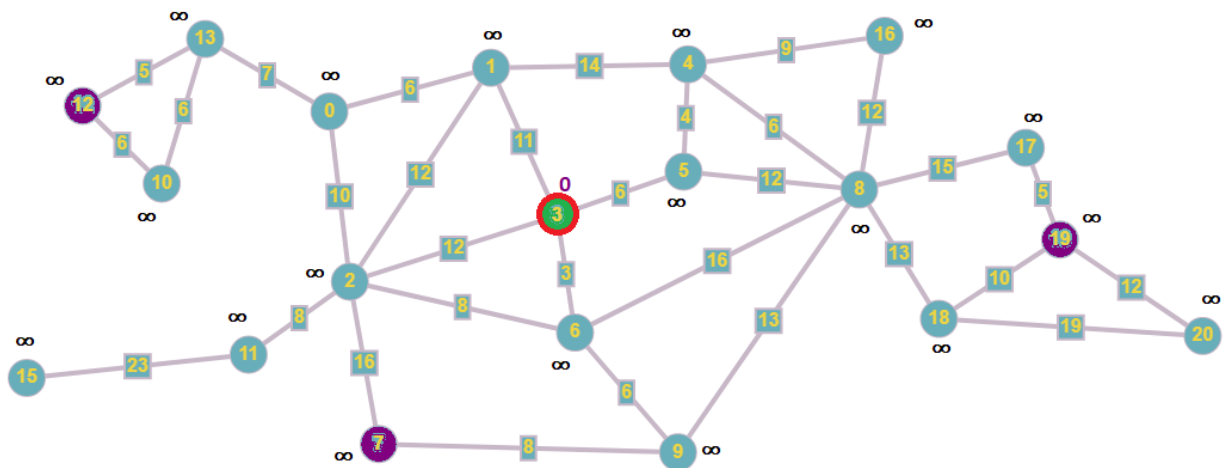


Рисунок 3.2 – Граф G після ініціалізації

**Крок 1**

Мінімальну відстань має вершина  $v_3$ , тому почнемо її обробляти.  
Суміжні їй вершини:  $v_1$ ,  $v_2$ ,  $v_5$  та  $v_6$ .

Рисунок 3.3 – Обираємо вершину  $v_3$  для опрацювання

По черзі намагаємось зменшити потенційну відстань до кожної з суміжних до  $v_3$  вершин. Якщо сума відстані до вершини, що опрацьовуємо, та довжини ребра менша за поточну потенційну відстань, то записуємо нову потенційну відстань до суміжної вершини.

Для  $v_1$ :  $0+11=11 < \infty$  — записуємо нову потенційну відстань —  $D_1=11$ .  
Аналогічно для  $v_2$ ,  $v_5$ ,  $v_6$ :  $D_2=12$ ,  $D_5=6$ ,  $D_6=3$ .

Проставимо відповідні мітки на рисунку 3.4.

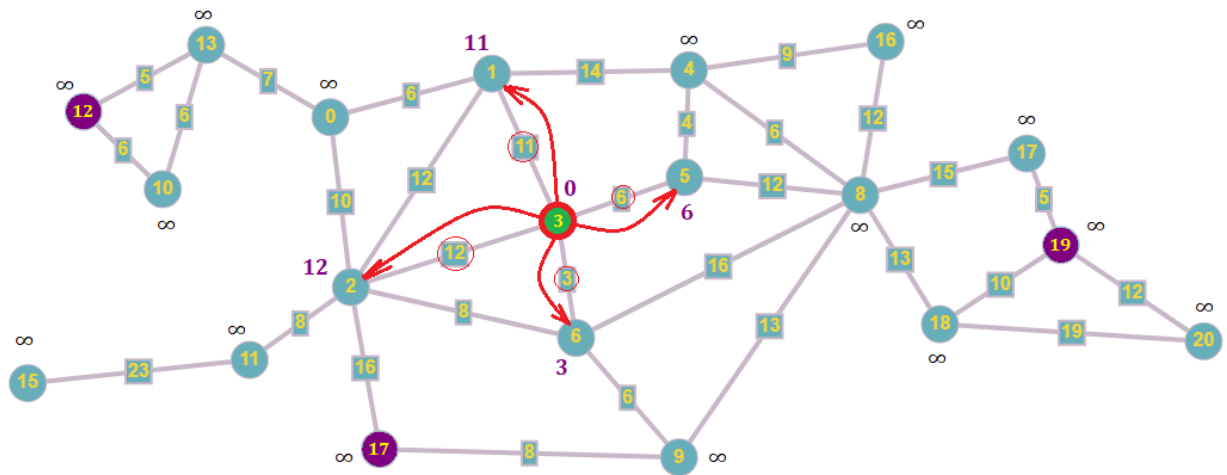


Рисунок 3.4 – Граф з новими потенційними відстанями для суміжних,  
до  $v_3$  вершин

Усі суміжні до  $v_3$  вершини перевірені. Поточну відстань до  $v_3$  тепер вважаємо остаточною ( $d_3=0$ ) та відмітимо  $v_3$  на графі (рисунок 3.5).

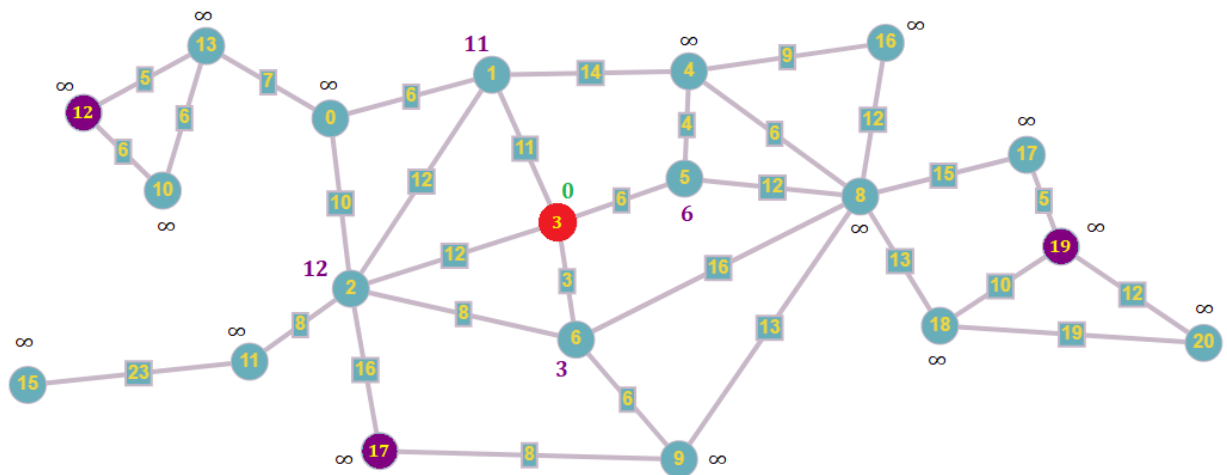
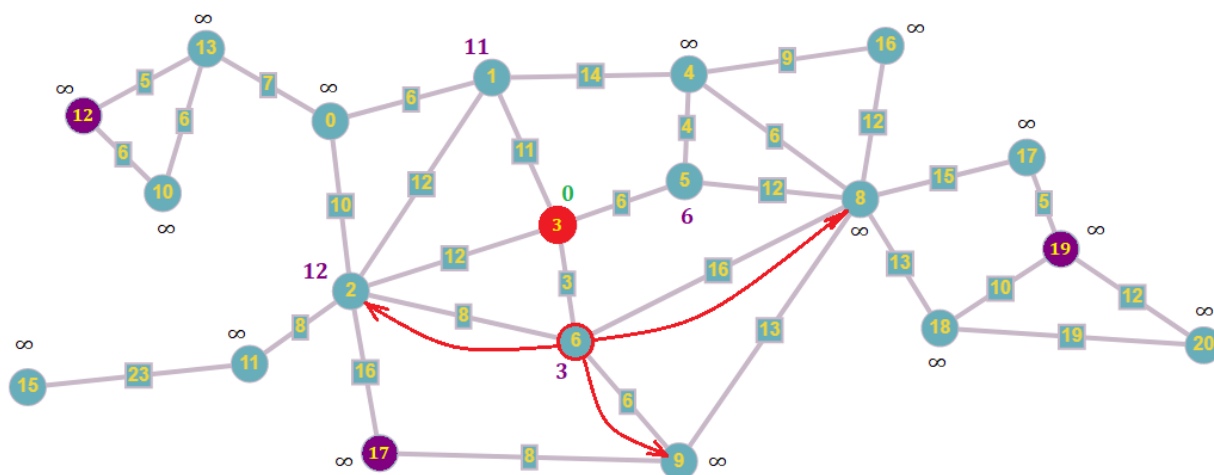


Рисунок 3.5 – Результат першого кроку

## Крок 2

Повторюємо крок алгоритму для вершини з найменшою потенційною відстанню —  $v_6$ .

Рисунок 3.6 – Обираємо вершину  $v_6$  для опрацювання

Перераховуємо потенційні відстані для суміжних до  $v_6$  вершин:

$D_2=12$ ,  $D_8=19$ ,  $D_9=9$ ;

Записуємо остаточну відстань  $d_6=3$ , та відмічаємо  $v_6$  як опрацьовану.

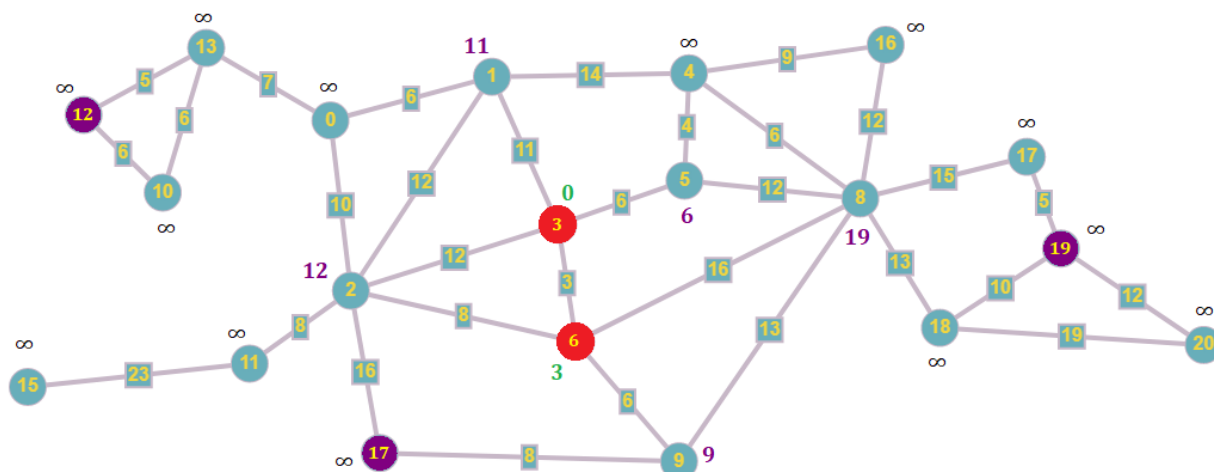


Рисунок 3.7 – Результат 2-го кроку

Аналогічно всі наступні кроки покажемо на рисунках 3.8-20.

## Крок 3

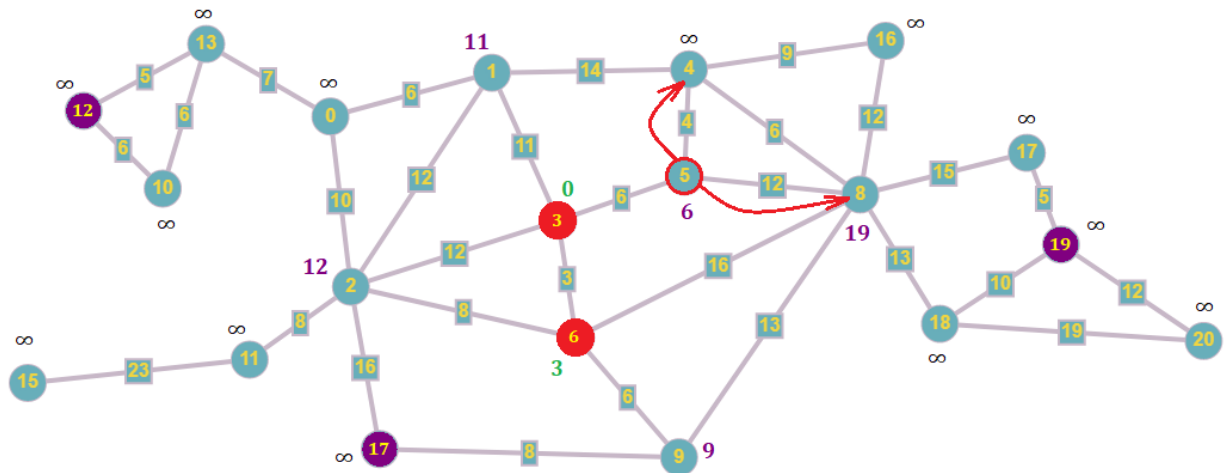
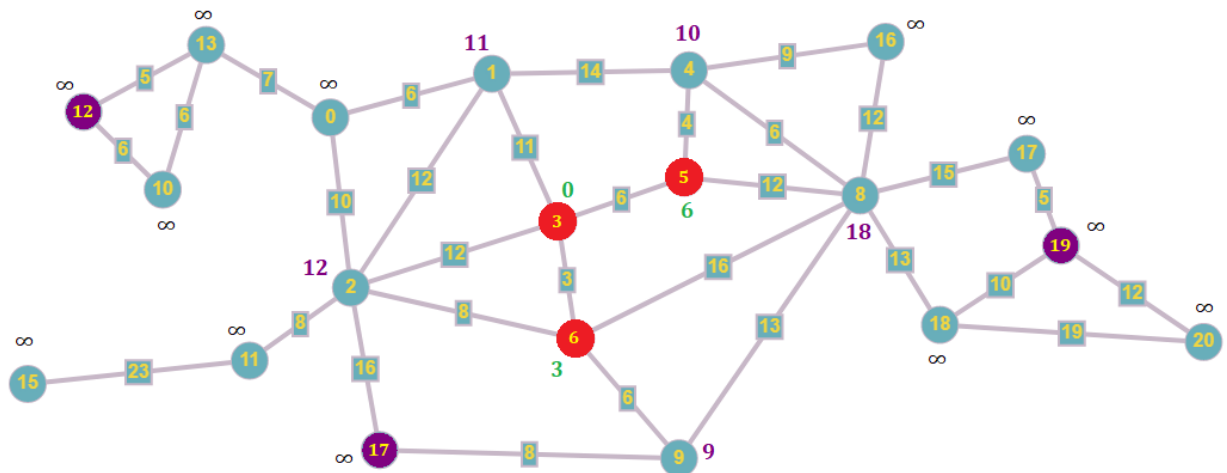
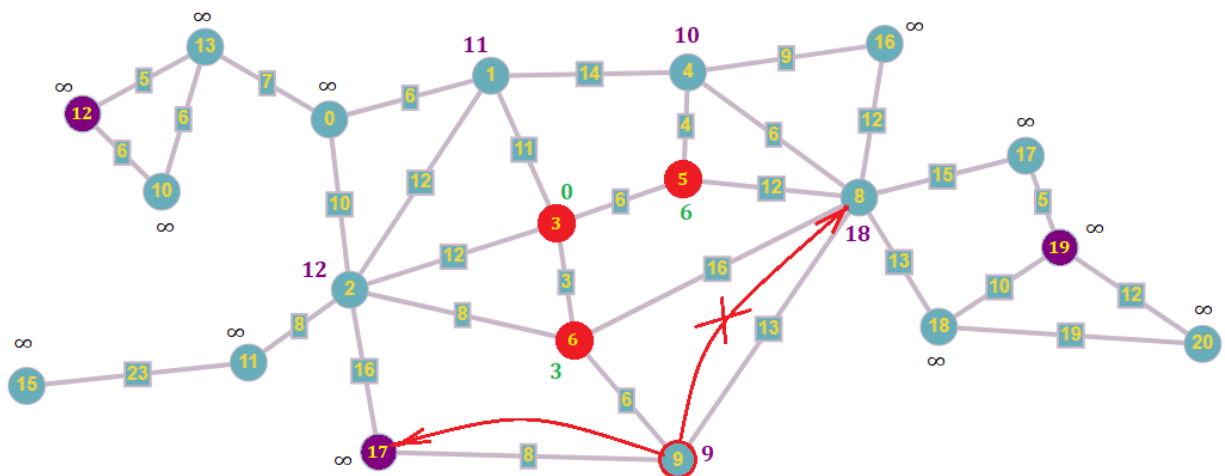
Рисунок 3.8 – Обираємо вершину  $v_5$  для опрацювання

Рисунок 3.9 – Результат 3-го кроку

## Крок 4

Рисунок 3.10 – Обираємо вершину  $v_9$  для опрацювання

Змн.	Арк.	№ докум.	Підпис	Дата

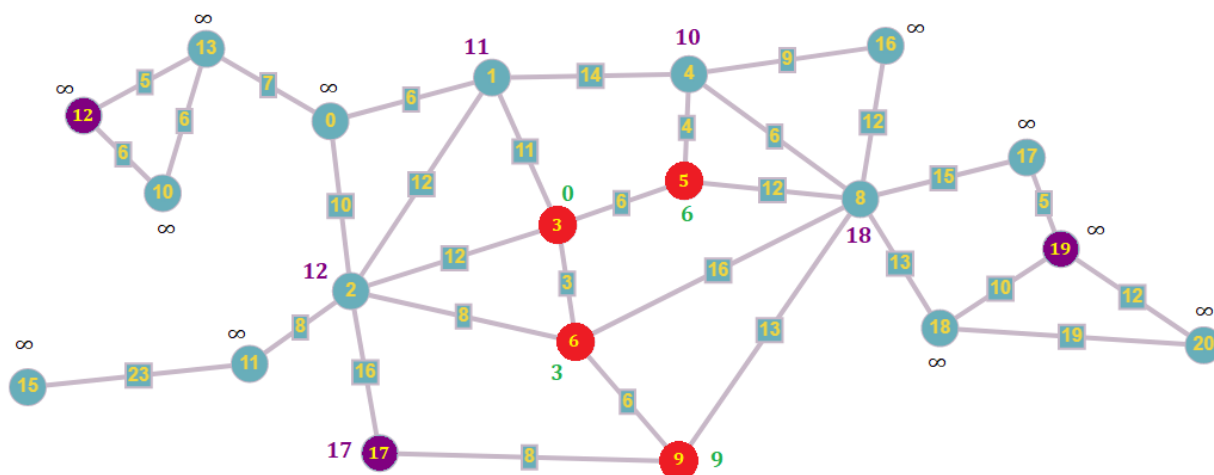


Рисунок 3.11 – Результат 4-го кроку

## Крок 5

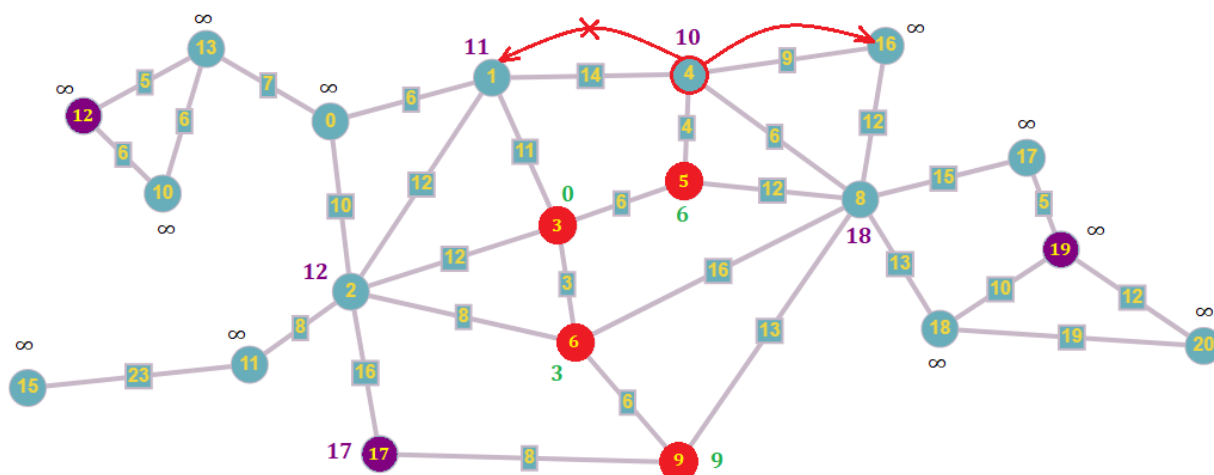
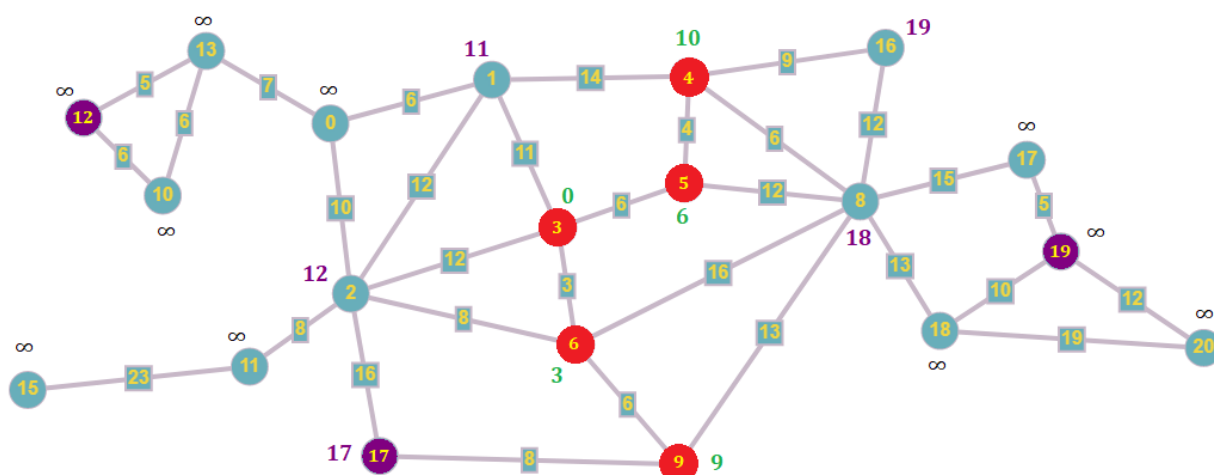
Рисунок 3.12 – Обираємо вершину  $v_4$  для опрацювання

Рисунок 3.13 – Результат 5-го кроку

Змн.	Арк.	№ докум.	Підпис	Дата

## Крок 6

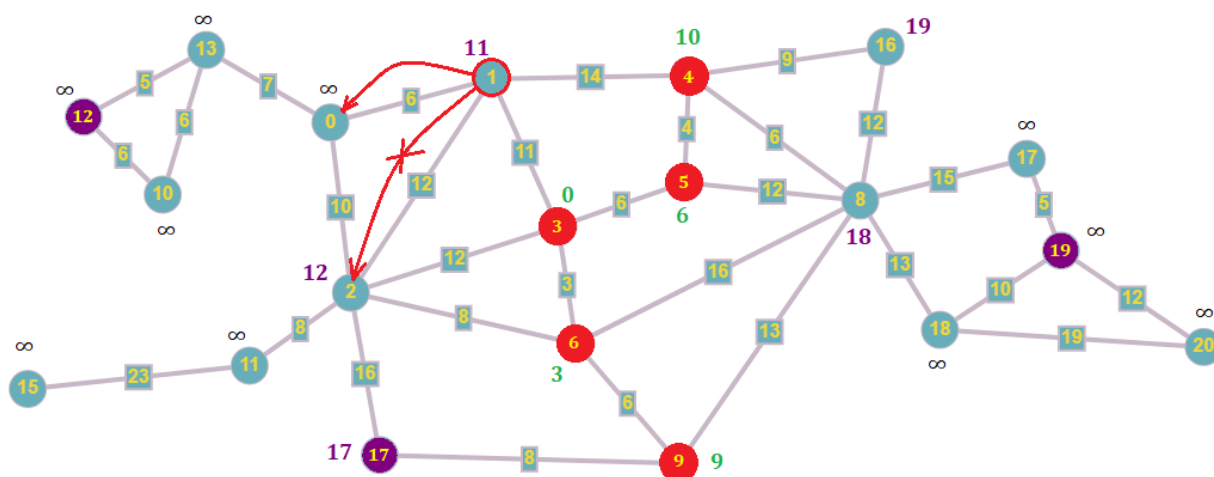
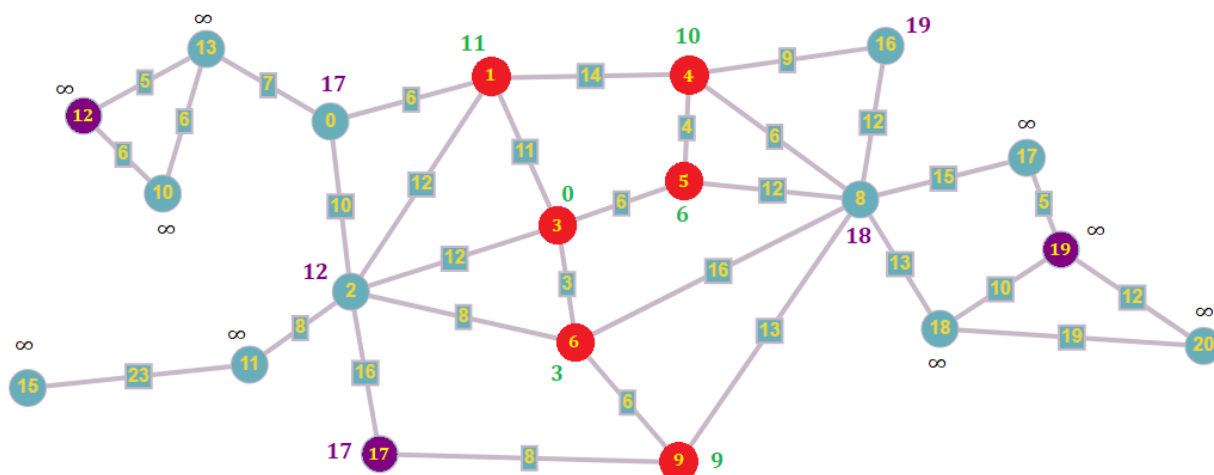
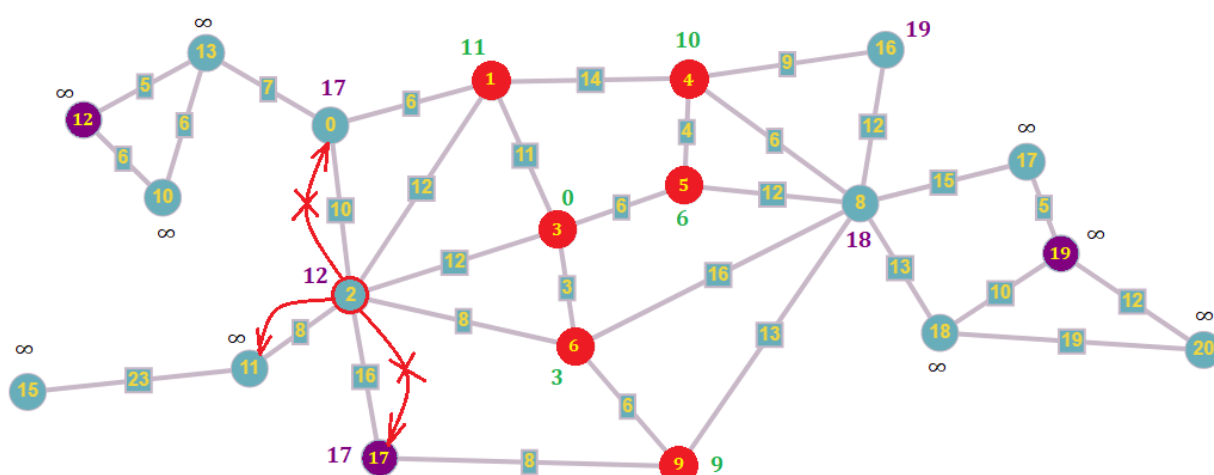
Рисунок 3.14 – Обираємо вершину  $v_1$  для опрацювання

Рисунок 3.15 – Результат 6-го кроку

## Крок 7

Рисунок 3.16 – Обираємо вершину  $v_2$  для опрацювання

Змн.	Арк.	№ докум.	Підпис	Дата



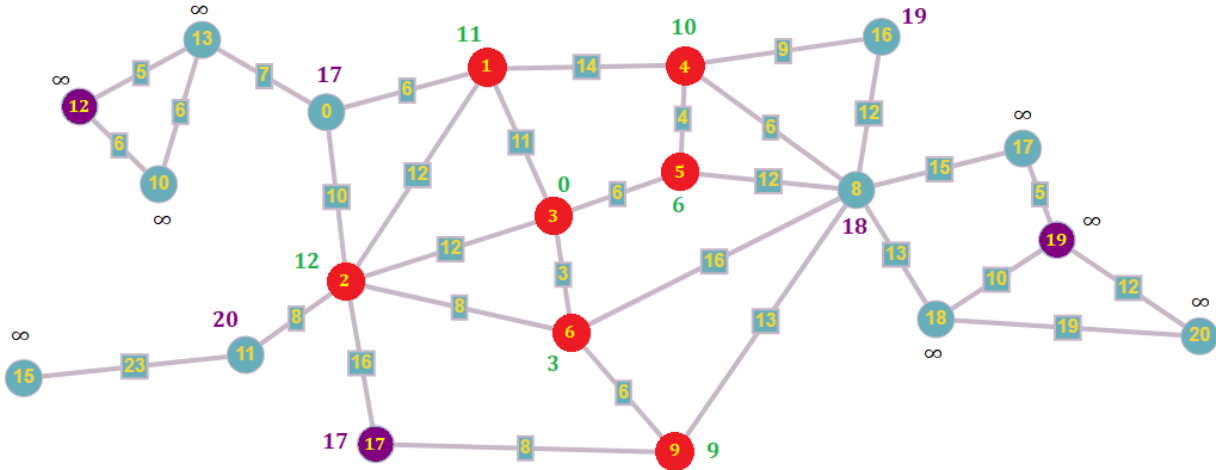


Рисунок 3.17 – Результат 7-го kroку

## Крок 8

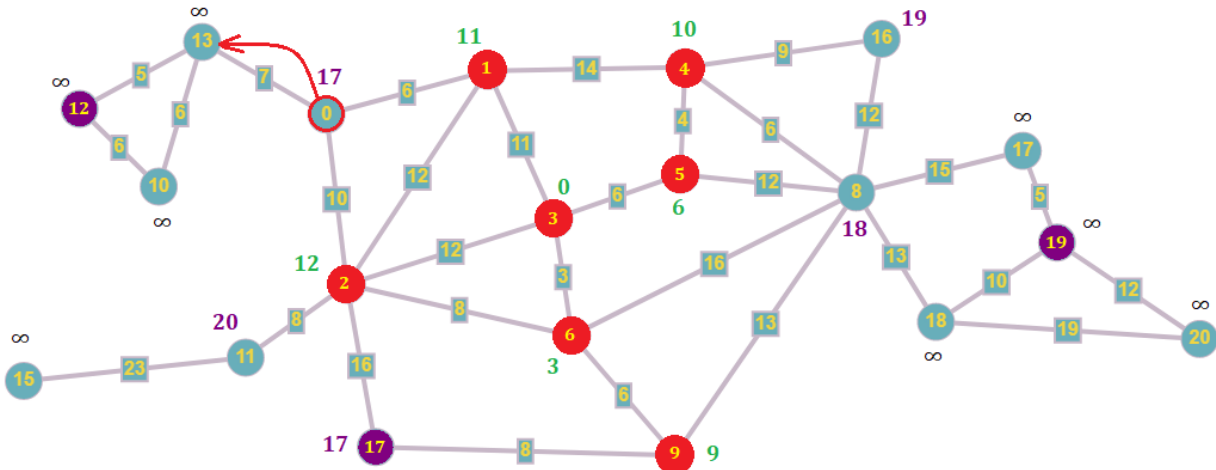


Рисунок 3.18 – Обираємо вершину  $v_0$  для опрацювання

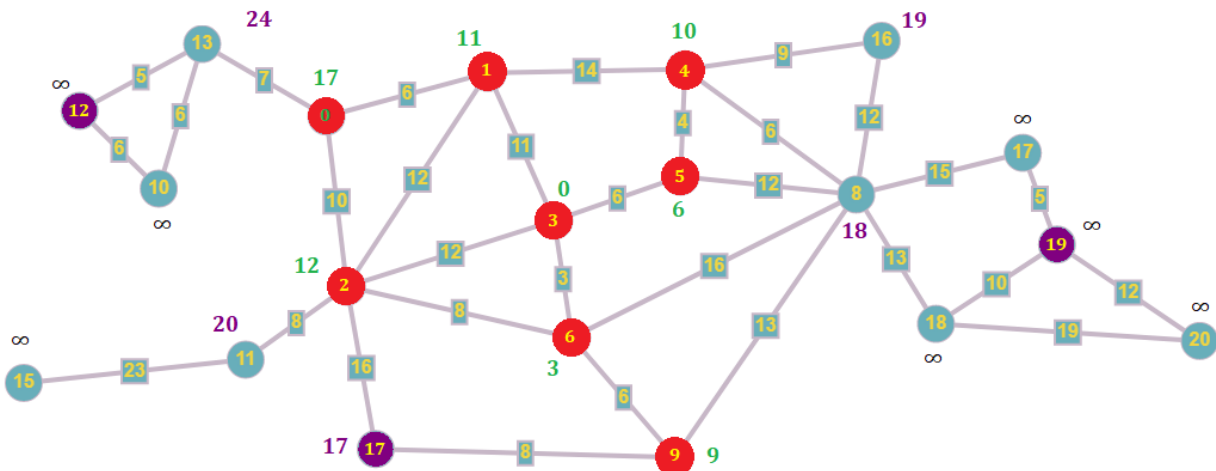


Рисунок 3.19 – Результат 8-го kroky

**Крок 9**

На опрацювання обрано вершину  $v_{17}$ , яка належить до кінцевих, а отже ми знайшли найближчу потрібну вершину та відстань до неї — 17.

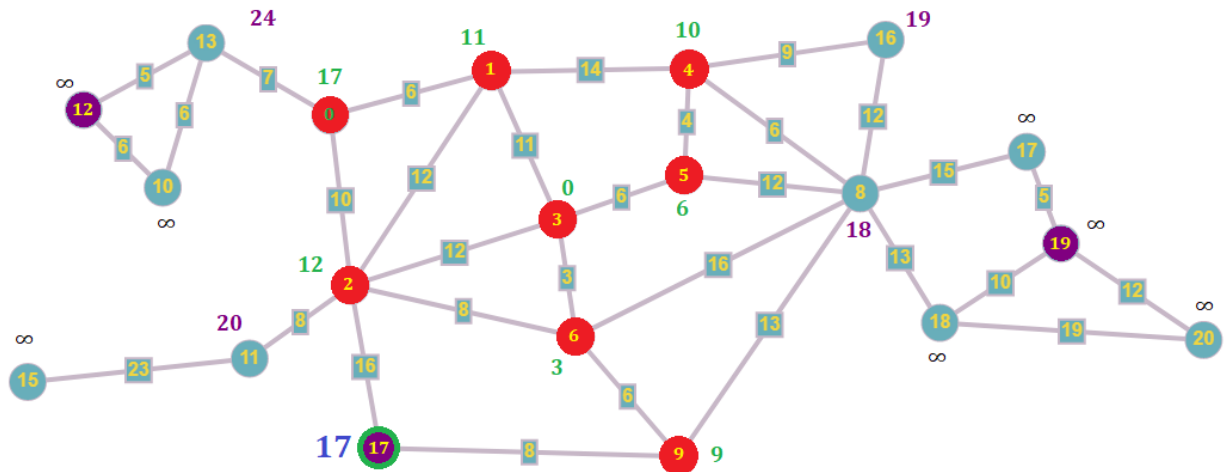


Рисунок 3.20 – Результат роботи алгоритму

**Висновок до розділу**

В розділі математичного забезпечення сформульовано змістовну та математичну постановку задачі пошуку найкоротшого шляху. Представлено математичне обґрунтування, а методом пошуку найкоротшого шляху було обрано алгоритм Дейкстри. Також було наведено детальний опис алгоритму розв'язання поставленої задачі та розглянуто приклад виконання алгоритму.

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Під час створення програмного продукту в якості IDE (інтегрованого середовища розробки) було обрано Android Studio[9]. Android Studio побудована на базі IntelliJ Idea[10] та є офіційним засобом розробки Android застосувань. Android Studio надає розробникам низку функцій, таких як: редактор макетів WYSIWYG, інтегрована система збірки Gradle, засоби рефакторингу коду, система тестування, емулятор Android пристроїв, вбудованна підтримка Google Cloud Platform, здатність роботи з оновленим компілятором Jack, а також покращена підтримка Java 8 і вдосконалена функція Instant Run.

В якості мови програмування було обрано Java — мову програмування, яка слідує парадигмі ООП. Java — вже не єдинина мова, що має офіційну підтримку для розробки на Android, тим паче Java залишається передовою актуальною мовою програмування, що має багато переваг, таких як:

- об'єктно орієнтоване програмування — концепція, в якій ви визначаєте тип даних і його структуру, а також набір функцій, застосовуваних до нього. Таким чином, структура даних стає об'єктом, яким можна управляти для створення відносин між різними об'єктами. При іншому ж підході - процедурному програмуванні - потрібно слідувати чітким інструкціям, використовувати змінні і функції. При ООП можна групувати ці змінні і функції за допомогою контексту, маркувати їх і посилатися на функції в контексті кожного конкретного об'єкта;

- Java має підтримку великих кількості бібліотек;
- безпека — по-перше, на відміну від C, в Java немає покажчиків, які можуть викликати несанкціонований доступ до пам'яті, а по-друге, в Java є Security Manager;

— ефективність — ефективність безпосередньо пов'язана з пам'яттю, обсяг якої обмежений. Розробникам Java не потрібно вручну писати код для управління пам'яттю завдяки автоматичному управлінню пам'яттю (АММ).

У якості бази даних було обрано SQLite[11] — бібліотеке на мові С, яка реалізує невеликий, швидкий, автономний, високо-, повнофункціональний механізм бази даних SQL. SQLite - найпоширеніший в світі движок баз даних, він вмонтований в усі мобільні телефони і більшість комп'ютерів і постачається у безлічі інших застосувань, які люди використовують кожен день. Так як для впровадження розробленого програмного продукту у вже існуючу систему екстреного реагування, потрібно буде реалізувати репозиторії для роботи з базою даних, яка в даний момент використовується у програмному забезпеченні екстрених служб, то вибір SQLite, як спосіб запису даних до внутрішньої бази даних та демонстрування реалізації репозиторіїв, буде найбільш доцільним.

В представленій роботі однією із задач є пошук найкоротшого шляху до місця надзвичайної ситуації. Для побудови таких маршрутів можна було б використати найбільш популярний та, напевне, найкрасивіший в плані UI варіант — Google Maps Api[12], який у поєднанні з Directions API[13] з пакету Google Web Services дозволяє зручно та швидко це зробити. У третьому розділі даної роботи був обранай алгоритм Дейкстри на графах. Але нажаль Google Maps не розповсюджують свої географічні дані вільно, тому завантажити їх та конвертувати у граф не представляється можливим. Але є на допомогу прийде проект OpenStreetMaps[14], який дозволяє експортувати дані мап та вільно використовувати їх під відкритою ліцензією. OpenStreetMaps — це відкритий проект, в якому можуть прийняти участь всі бажаючі волонтери. Хоча на основі OSM побудовано багато проектів, інструментів та різних доповнень, OpenStreetMaps не можуть відзначитись зручністю та зовнішнім виглядом своїх мап. Але велика кількість

стороннього інструментарію дає дуже гнучкі умови для експортування географічних даних. Одним з таких прикладів є надзвичайно зручна уліта Overpass Turbo[15], створена для фільтрації OpenStreetMaps. Вона використовує власну мову запитів, що дозволяє створювати складні запити для експортування даних з OSM за власними критеріями та в зручному форматі. Дякуючи активному користувачу habr.com — Anvaka — експортування транспортних графів можна зовсім автоматизувати за допомогою використання скриптів, що користувач виклав на github[16]. В результаті отримаємо транспортний граф потрібного міста у форматі JSON, з якого легко сформувати граф завдяки Gson[17] — бібліотеці для Java від Google, що дозволяє конвертувати JSON у Java об'єкти. Завдяки цим маніпулюванням ми отримаємо можливість виконувати пошук найкоротшого маршруту, використовуючи алгоритм Дейкстри на отриманому графі, а відображати маршрут на карті від Google.

Також у нашу систему потрібно інтегрувати лайв чат для комунікації з диспетчером після того, як користувач здійснить виклик. В нашому випадку найзручніше це зробити буде за допомогою FirebaseUI[18] від Firebase[19] — дочірньої компанії Google. По-перше, це дуже зручний швидкий спосіб створити чат, а по-друге, в нашому проекті ми вже маємо гугл сервіси, що потребували підключення Firebase, що допоможе нам відмовитись від захламлення gradle'у проекту новими dependencies, а ми будемо користуватись добре синхронізованими продуктами, керувати та відслідковувати які можна в одному місці — в консолі Firebase.

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Для забезпечення правильної та стабільної роботи даного програмного продукту висуваються такі вимоги:

- смартфон на базі ОС Android;
- версія Android 4.0(Ice Cream Sandwich) або вища.

## 4.3 Архітектура програмного забезпечення

Для побудови правильної архітектури програмного продукту потрібно вирішити який архітектурний шаблон обрати. Існує досить багато шаблонів для архітектур застосувань. Найбільш відомими є класичні трирівневі архітектури, такі як:

- MVC: Model-View-Controller
- MVP: Model-View-Presenter
- MVVM: Model-View-ViewModel

Всі ці шаблони представляють основну подібну ідею - структурувати код вашого проекту таким чином, щоб він був розділений різними загальними шарами. Кожен шар має свою відповідальність. Проект стає модульним: окремі частини коду більш придатні для тестування, а застосування досить гнучким для безперервних змін.

Розглянемо кожний шаблон по черзі.

### Model-View-Controller (MVC)

Цей шаблон був першим кроком в розробці правильної архітектури Android-застосувань. Він пропонує розділити ваш код на 3 різних рівня:

Model — шар даних. Відповідає за управління бізнес-логікою і взаємодію з мережею і рівнями бази даних.

View — рівень інтерфейсу користувача. Це проста візуалізація даних з моделі.

Controller — логічний рівень, отримує повідомлення про поведінку користувача і оновлює модель в міру необхідності.

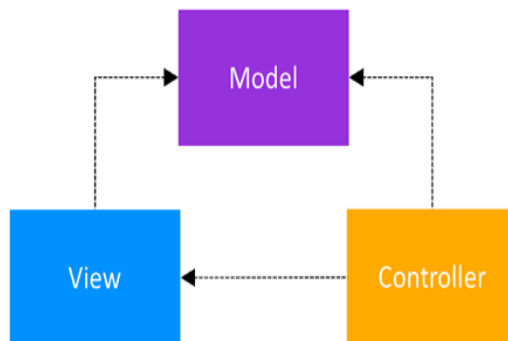


Рисунок 4.1 – Схема структурна шаблону MVC

На рисунку 4.1 представлена схема MVC. На ній ми бачимо, що і контролер, і представлення залежать від моделі. Контролер оновлює дані. Представлення отримує дані. Проте, модель відокремлена і може бути протестована незалежно від призначеного для користувача інтерфейсу.

При застосуванні MVC до проекту Android виявляється, що шари коду залежить один від одного. Саме цьому MVC зазвичай не використовується при розробці під Android.

### **Model-View-Presenter (MVP)**

Після першого підходу, який не спрацював, розробники Android пішли далі і спробували використовувати один з найпопулярніших архітектурних шаблонів - MVP. Цей шаблон став широко використовуватися і до сих пір рекомендується для впровадження. Розглянемо ролі його 3-ох окремих шарів:

Model — шар даних, такий самий, як у шаблоні MVC. Відповідає за управління бізнес-логікою і взаємодія з мережею і рівнями бази даних.

View — рівень інтерфейсу користувача. Відображає дані і повідомляє презентер про дії користувача.

Presenter — отримує дані з моделі, застосовує логіку призначену для інтерфейсу користувача і управляє станом уявлення, вирішує, що

відображати, і реагує на повідомлення введення користувачем з представлення. По суті, це контролер MVC, за винятком того, що він взагалі не прив'язаний до View, а лише до інтерфейсу.

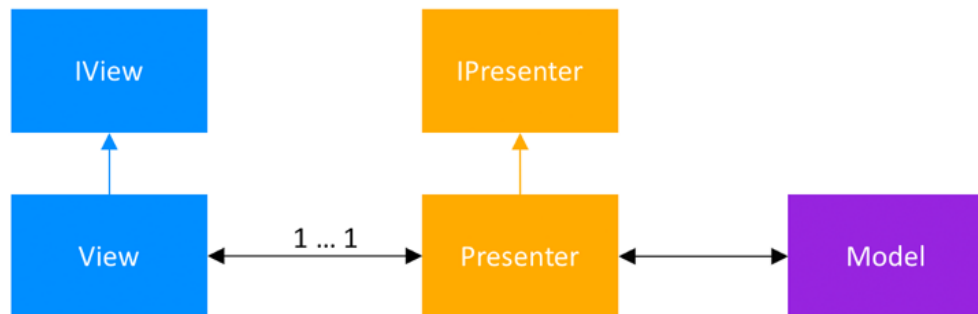


Рисунок 4.2 – Схема структурна шаблону MVP

Схема MVP показує, що View і Presenter тісно пов'язані. Вони повинні мати посилання один на одного. Їхні стосунки визначені в Contract класі інтерфейсу.

Ця модель має один істотний, але контрольований недолік. Presenter має тенденцію розширюватися до величезного всезнаючого класу, якщо ви не розділяєте свій код відповідно до принципу єдиної відповідальності. Однак, модель MVP пропонує дуже гарний поділ інтересів.

### Model-View-ViewModel (MVVM)

Патерн MVVM — це третій крок розробки підходу до побудови архітектури. Він став шаблоном архітектури, рекомендованим командою Android з випуском компонентів архітектури Android. Розглянемо окремі шари коду:

Model — абстрагує джерело даних. ViewModel працює з моделлю для отримання і збереження даних.

View — інформує ViewModel про дії користувачів.

ViewModel — надає потоки даних, що відносяться до представлення.

Різниця в порівнянні з шаблоном MVP полягає в тому, що в MVVM ViewModel не містить посилання на уявлення, як в Presenter. У MVVM ViewModel надає потік подій, до якого можуть прив'язуватися різні



представлення. З іншого боку, в разі MVP Presenter безпосередньо повідомляє поданням, що відображати. Давайте подивимося на схему MVVM на рисунку 4.3.



Рисунок 4.3 – Схема структурна шаблону MVP

У MVVM View має посилання на ViewModel. ViewModel не має інформації про подання. Між View і ViewModel існує відношення багато-до-одного.

При побудові архітектури нашого програмного продукту був обраний архітектурний шаблон MVVM (Model-View-ViewModel) з використанням технології зв'язування даних (Data Bindings), основною концепцією якого є відділення логіки представлення даних від бізнес-логіки шляхом винесення її в окремий клас для більш чіткого розмежування.

#### 4.3.1 Діаграма класів

Структурна схема класів наведена в графічному матеріалі.

За концепцією MVVM представлені класи можна поділити на Model, View та ViewModel.

До шару представлення відносяться всі класи фрагментів та активіті (а також XML-файли, що зберігають макет представлення): CallFragment, CallProcessFragment, WaitingFragment, ChatFrgment, ManualOneFragment, MapActivity, ManualFragment, SosSettingsFragment, MedicalRecordFragment.

До шару моделі відносяться всі репозиторії для роботи з базою даних і арі та класи логіки оброблення цих даних: UserRepository, SituationTypeRepository, ResultRepository, ChatRepository, ChatMessageModel, Dijkstra, Graph, Node, Edge, Route, LocationLatLng, Result, User, MedicalRecord.

До шару ViewModel відносяться всі класи, що працюють з репозиторіями та надають дані шару представлення: CallSharedViewModel, CallProcessViewModel, ChatViewModel, ManualOneViewModel, MapViewModel, ManualViewModel, SosSettingsViewModel, MedicalRecordViewModel.

#### 4.3.2 Діаграма послідовності

Структурна схема послідовності наведена в графічному матеріалі.  
На представленій схемі зображено процес виконання виклику.

#### 4.3.3 Діаграма компонентів

Структурна схема компонентів наведена в графічному матеріалі.

#### 4.3.4 Специфікація функцій

Основні функції описані у таблиці 4.1.

Таблиця 4.1 — Специфікація функцій

Назва	Опис
Клас Fragment – реалізує поведінку або частину користувацького інтерфейсу. Функції, описані для цього класу, є загальними для усіх наступних класів із закінченням Fragment у назві, так як вони є нащадками даного класу	
onCreate(Bundle savedInstanceState)	Первісне створення фрагменту
onCreateView()	Виконується після onCreate() для створення інтерфейсу користувача
onStart()	Виконується після onCreateView() коли фрагмент стає видимий для користувача.
onResume()	Виконується після onPause() для початку інтеракції з користувачем.

Продовження таблиці 4.1

Назва	Опис
onStop()	Виконується, коли фрагмент більше не видимий для користувача.
CallSharedViewModel — ViewModel для обміну даними між CallFragment та CallProcessFragment	
select(int emergency)	CallFragment передає інформацію про обрану екстрену службу.
getSelected()	CallProcessFragment отримує інформацію про обрану екстрену службу.
CallProcessViewModel — надає дані CallProcessFragment	
addPhoto(Uri uri)	Викликається CallProcessFragment при додаванні фото
delPhoto(int pos)	Викликається CallProcessFragment при видаленні фото
addInjured(String phone)	Викликається CallProcessFragment при додаванні постраждалого, працює з UserRopository для пошуку користувача по номеру телефона
delInjured(int pos)	Викликається CallProcessFragment при видаленні постраждалого
getSpinnerAdapter(int emergency)	Викликається CallProcessFragment при заповненні Spinner вибору типу ситуації, працює з SituationTypeRepository для отримання списку типів
getPhotosAdapter()	Викликається CallProcessFragment при заповненні RecyclerView фото

## Продовження таблиці 4.1

Назва	Опис
getInjuredAdapter()	Викликається CallProcessFragment при заповненні RecyclerView постраждалих
setSituationType(String type)	Викликається CallProcessFragment при виборі типу ситуації
setInfo(String text)	Викликається CallProcessFragment при заповненні інформації про виклик
sosCall()	Викликається CallProcessFragment при натисканні на кнопку SOS, передає данні ResultRepository для збереження
saveResult()	Завершує виклик, передає данні ResultRepository для збереження
ChatViewModel — надає дані ChatFrgment	
sendMessage(String text)	Передає повідомлення ChatRepository
getDisplayMessagesAdapter(Fragment fragment)	Отримує повідомлення від ChatRepository
ManualOneViewModel— надає дані ManualOneFrgment	
getManualId(situationTypeId int)	Повертає id html-файлу ресурсів для відображення у HtmlTextView
MapActivity — відображення та робота з картою	
displayRoute(List<LatLng>)	Відрисовує маршрут на карті
MapViewModel — надає данні MapActivity	
getResultString()	Повертає результат у вигляді рядку, отримує данні з ResultRepository
getRoute()	Повертає маршрут у вигляді спитку LatLng, отримує данні з ResultRepository
ManualViewModel — надає дані ManualFrgment	

Продовження таблиці 4.1

Назва	Опис
getTitlesAdapter()	Викликається ManualFrgment при заповненні RecyclerView заголовків
SosSettingsViewModel— надє дані SosSettingsFrgment	
getContactsAdapter()	Викликається SosSettingsFrgment при заповненні RecyclerView екстрених контактів
addContact(String number)	Викликається SosSettingsFrgment при додаванні екстреного контакту
delContact(int pos)	Викликається SosSettingsFrgment при видаленні екстреного контакту
setMessage(String text)	Викликається SosSettingsFrgment при додаванні повідомлення
change101()	Змінює значення необхідності виклику відповідної екстреної служби
change102()	Змінює значення необхідності виклику відповідної екстреної служби
change103()	Змінює значення необхідності виклику відповідної екстреної служби
change104()	Змінює значення необхідності виклику відповідної екстреної служби
saveSettings()	Зберігає налаштування у SharedPreferences
MedicalRecordViewModel— надє дані MedicalRecordFrgment	

Продовження таблиці 4.1

Назва	Опис
saveMedRecord(String diseases, String allergicReactions, String medicines, String bloodType, String growth, String weight)	Передає данні UserRepository для збереження
Dijkstra — виконання алгоритму Дейкстри	
findFire(int startId)	Пошук найближчої пожежної частини
findPolice(int startId)	Пошук найближчої поліцейської частини
findMed(int startId)	Пошук найближчої лікарні
findGas(int startId)	Пошук найближчої частини служби газу
Route — маршрут	
getLocationsList()	Повертає маршрут у вигляді списку LocationLatLng
getLatLngList()	Повертає маршрут у вигляді списку LatLng
getString()	Повертає маршрут у вигляді рядку
LocationLatLng — розміщення Node графу на карті по довготі і широті	
toLatLng()	Повертає значення у вигляді LatLng
ResultRepository — працює з базою даних SQLite у контексті результатів виклику	
addResult(Result result)	Обробляє та зберігає результат виклику
getResult(int id)	Повертає результат з бази даних
getResultRoute(int id)	Повертає маршрут з бази даних
UserRepository — працює з базою даних SQLite у контексті користувача та його медичної картки	
getUser(int id)	Повертає користувача з бази даних
setUser(int id, User user)	Оновлює користувача у базі даних

Продовження таблиці 4.1

Назва	Опис
addUser(User user)	Додає користувача до бази даних
delUser(int id)	Видаляє користувача з бази даних
getUserIdByPhone(String phone)	Повертає id користувача по його номеру телефона
getMedRec(int userId)	Повертає медичну картку користувача з бази даних
setMedRec(int userId, MedicalRecord medrec)	Оновлює медичну картку користувача в базі даних
SituationTypeRepository — працює з базою даних SQLite у контексті типів ситуацій	
getTypesList(int emergency)	Повертає список типів ситуацій
ChatRepository — працює з Firebase API для отримання та виправлення повідомлень	
displayMessages(Fragment fragment, int messageLayout, int messageTextLayoutId, int messageUserLayoutId, int messageTimeLayoutId)	Отримує повідомлення з бази даних Firebase
sendMessage(String message)	Відправляє повідомлення до бази даних Firebase
ChatMessageModel — модель роботи чату	
getMessageText()	Повертає текст повідомлення
setMessageText(String messageText)	Додає текст повідомлення
getMessageUser()	Повертає власника повідомлення

## Продовження таблиці 4.1

Назва	Опис
setMessageUser(String messageUser)	Додає власника повідомлення
getMessageTime()	Повертає час повідомлення
getMessageTime(long messageTime)	Додає час повідомлення

**Висновок до розділу**

В даному розділі детально описано програмне та технічне забезпечення системи швидкого реагування на екстрені ситуації.

Сформульовані вимоги до технічного забезпечення для коректної роботи системи.

Аргументовано вибір архітектурного патерну для створеного програмного забезпечення. За допомогою діаграм класів, послідовності та компонентів розглянуто архітектуру програмного забезпечення, описані основні функції



## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

Після встановлення застосування користувач потрапляє на сторінку авторизації (рисунок 5.2).

Користувач натискає на кнопку «Зареєструватись», потрапляє на сторінку реєстрації (рисунок 5.1) та заповнює поля:

- прізвище;
- ім'я;
- по-батькові;
- дата народження;
- місто;
- місце проживання;
- номер телефону;
- електронна пошта;
- пароль(пароль повторно).

The screenshot shows a mobile application interface for registration. At the top, there is a red header with the title 'Регістрація'. Below the header, the word 'Регістрація' is repeated in a smaller font. The form consists of several input fields, each preceded by a label: 'Прізвище:', 'Ім'я:', 'По-батькові:', 'Дата народження:', 'Місто:', 'Адреса проживання:', 'Номер телефону:', and 'E-mail:'. Each label is followed by a text input field containing a placeholder with the same text as the label.

Рисунок 5.1 – Сторінка реєстрації

Після натискання кнопки «Зареєструватись» внизу сторінки, користувач повертається на сторінку авторизації та заповнює електронну адресу і пароль, після чого натискає на кнопку «Увійти»:

Рисунок 5.2 – Сторінка реєстрації

Після авторизації користувач потрапляє на головну сторінку—сторінку виклику екстрених служб (Рисунок 5.3). Тут він може обрати одну з служб, щоб зробити виклик через застосування, або натиснути на кнопку виклику по телефону в звичайному режимі. Також в разі гострої необхідності користувач може натиснути на кнопку SOS.

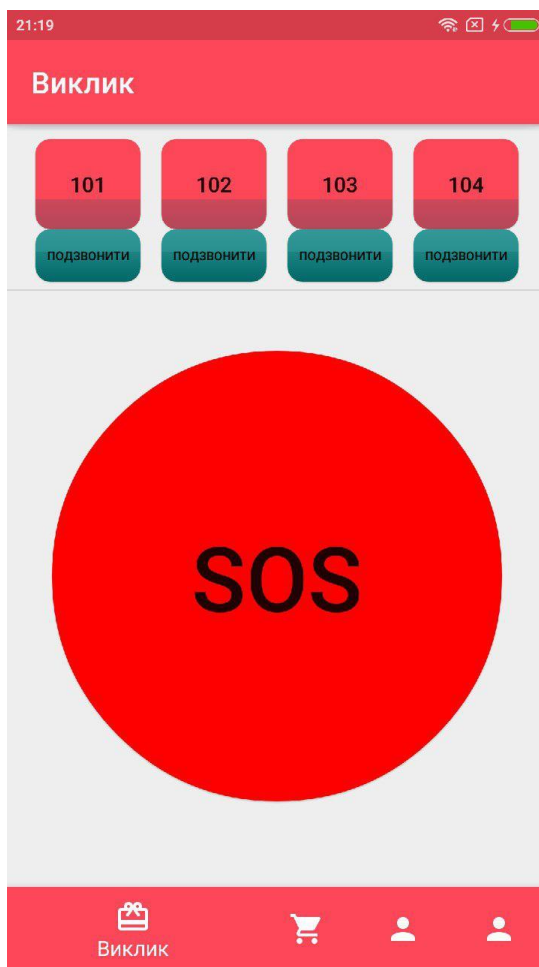


Рисунок 5.3 – Сторінка виклику екстрених служб

Обрав одну із екстрених служб для виклику через застосування, користувач починає процес виклику, який ми бачимо на рисунку 5.4. Користувач обирає тип ситуації, набирає текст, додає медіа файли та може додати постраждалих. Коли виклик заповнений, натискає кнопку «Викликати».

Рисунок 5.4 – Сторінка заповнення виклику

Після того, як виклик було здійснено, користувач переходить на сторінку очікування (рисунок 5.5), на якій йому виводиться довідкова інформація по його типу ситуації. Вище довідкової інформації йому доступний чат диспетчером.

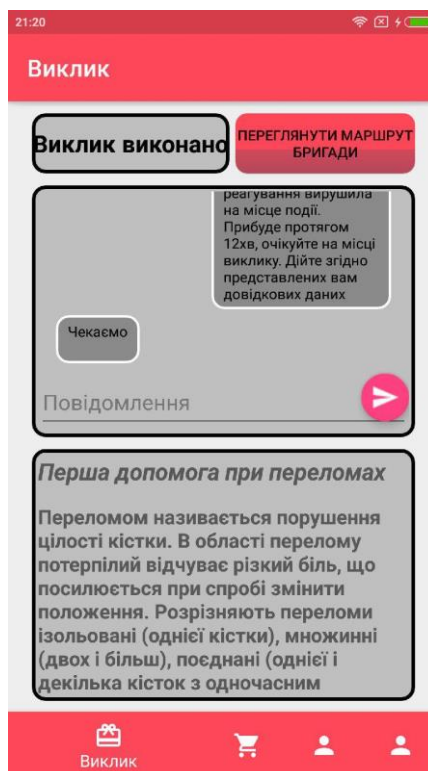


Рисунок 5.5 – Сторінка очікування

Також користувачу доступна кнопка «Показати маршрут бригади», яка покаже маршрут бригади екстреного реагування на карті, а також (в тестовому варіанті застосування) виведе повну інформацію про зроблений ВИКЛИК.

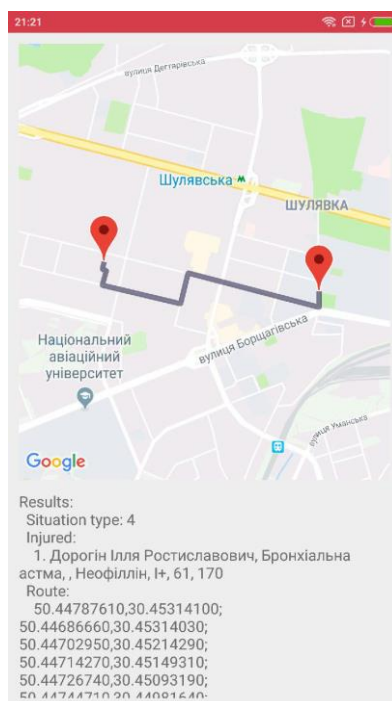


Рисунок 5.6 – Сторінка з картою

					ДП ІС-5208.1181-с.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

Також, користувачу доступна нижня панель навігації.

На другій сторінці можна отримати довідникову інформацію по різних типах ситуацій (рисунок 5.7). На цій сторінці доступний пошук. Натиснувши елемент списку, з'явиться відповідна інформація.

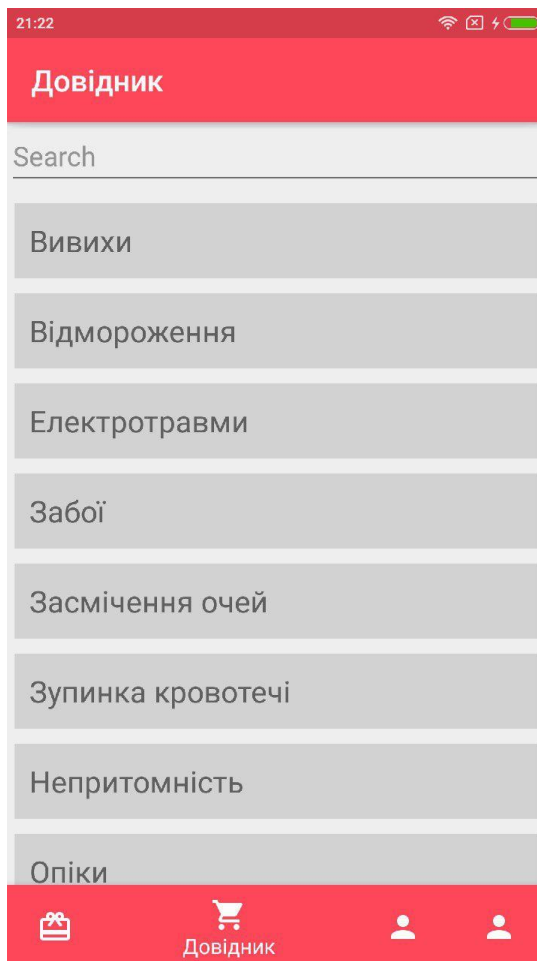


Рисунок 5.7 – Сторінка довідникових даних

Перейшовши по третьому пункту меню, користувач потрапляє до особистих налаштувань кнопки SOS (рисунок 5.8). Він може обрати для своєї кнопки SOS власний сценарій (якщо, наприклад, людина має захворювання на кшталт епілепсії вона заздалегідь знає найбільш ймовірний випадок натискання кнопки). Користувач набирає потрібне повідомлення та обирає екстрені служби і номери, до яких це повідомлення буде відправлено.

Рисунок 5.8 – Сторінка налаштувань кнопки SOS

На останій вкладці користувач може змінити дані своєї медичної картки, які будуть направлені до екстрених служб, в разі якщо він буде у списку постраждалих під час виклику.

Рисунок 5.9 – Сторінка особистої медичної картки

## 5.2 Випробування програмного продукту

### 5.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій комплексу задач автоматизованої системи швидкого реагування на екстрені ситуації вимогам технічного завдання.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

Після проведення тестування було перевірено всю функціональність системи. В представлених таблицях наведено результати випробувань основних функцій програмного продукту.

Таблиця 5.1 – Реєстрація нового користувача

Мета тесту	Перевірка функції «Реєстрація»
Початковий стан моделі	Відкрита сторінка реєстрації.
Вхідні дані:	Прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль.
Схема проведення тесту:	Заповнити наступні поля: прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль. Натиснути на кнопку «Реєстрація».
Очікуваний результат:	Реєстрація пройдена успішно, відкрита сторінка авторизації..



Продовження таблиці 5.1

Мета тесту	Перевірка функції «Реєстрація»
Стан моделі після проведення випробувань:	Реєстрація пройдена успішно, відкрита сторінка авторизації..

Таблиця 5.2 – Перевірка заповнення всіх полів при реєстрації

Мета тесту	Перевірка функції «Перевірка заповнення усіх полів при реєстрації»
Початковий стан моделі	Відкрита сторінка реєстрації
Вхідні дані:	Прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль за винятком деяких
Схема проведення тесту:	Заповнити наступні поля, залишивши деякі з них незаповненими: прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль.
Очікуваний результат:	Незаповнені поля підсвічуються червоним кольором
Стан моделі після проведення випробувань:	Незаповнені поля підсвічуються червоним кольором

Таблиця 5.3 – Вхід користувача у систему

Мета тесту	Перевірка функції «Авторизація»
Початковий стан моделі	Відкрита сторінка авторизації
Вхідні дані:	Електронна пошта та пароль користувача
Схема проведення тесту:	Заповнити поля електронної пошти та пароллю. Натиснути на кнопку «Вхід».
Очікуваний результат:	Відкрита сторінка виклику екстрених служб.
Стан моделі після проведення випробувань:	Відкрита сторінка виклику екстрених служб.

Таблиця 5.4 – Перевірка правильності введених даних

Мета тесту	Перевірка функції «Перевірка правильності введених даних»
Початковий стан моделі	Відкрита сторінка авторизації
Вхідні дані:	Електронна пошта та пароль користувача
Схема проведення тесту:	При заповненні полів електронної пошти та паролю, ввести неправильні дані. Натиснути на кнопку «Вхід».
Очікуваний результат:	Повідомлення «Введено некоректні дані».
Стан моделі після проведення випробувань:	Повідомлення «Введено некоректні дані».

Таблиця 5.5 – Вибір екстренної служби

Мета тесту	Перевірка функції «Вибір екстренної служби»
Початковий стан моделі	Відкрита головна сторінка застосування.
Вхідні дані:	
Схема проведення тесту:	Натиснути на кнопку однієї з екстрених служб
Очікуваний результат:	Відкрита сторінка виклику обраної екстреної служби.
Стан моделі після проведення випробувань:	Відкрита сторінка виклику обраної екстреної служби.

Таблиця 5.6 – Виклик екстренної служби

Мета тесту	Перевірка функції «Виклик екстренної служби»
Початковий стан моделі	Відкрита сторінка виклику екстренної служби
Вхідні дані:	Тип ситуації, додаткова інформація, постраждали.
Схема проведення тесту:	Користувач з випадного списку обирає потрібну ситуацію, заповнює поле «додаткова інформація» та додає користувачів до списку постраждалих.
Очікуваний результат:	Відкрита сторінка очікування з довідковою інформацією по обраному типу ситуації та чатом з диспетчером.
Стан моделі після проведення випробувань:	Відкрита сторінка очікування з довідковою інформацією по обраному типу ситуації та чатом з диспетчером.

Таблиця 5.7 – Перегляд маршруту бригади

Мета тесту	Перевірка функції «Перегляд маршрути бригади»
Початковий стан моделі	Відкрита сторінка очікування
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Маршрут бригади».
Очікуваний результат:	Відкрита сторінка з мапою, на якій відображено маршрут бригади до місця виклику
Стан моделі після проведення випробувань:	Відкрита сторінка з мапою, на якій відображено маршрут бригади до місця виклику

Таблиця 5.8 – Створення медичної картки

Мета тесту	Перевірка функції «Створення медичної картки»
Початковий стан моделі	Відкрита сторінка медичної картки
Вхідні дані:	Хвороби, алергічні реакції, медикаменти, група крові, вага, зріст.
Схема проведення тесту:	Користувач заповнює доступні поля поля
Очікуваний результат:	Показана спливаюча підказка про збереження медичної картки. Оновлена медична картка.
Стан моделі після проведення випробувань:	Показана спливаюча підказка про збереження медичної картки. Оновлена медична картка.

Таблиця 5.9 – Налаштування кнопки SOS

Мета тесту	Перевірка функції «Налаштування кнопки SOS»
Початковий стан моделі	Відкрита сторінка налаштувань кнопки SOS
Вхідні дані:	Екстренні служби, екстрені контакти, примітка
Схема проведення тесту:	Користувач обирає потрібні екстренні служби та контакти, до яких буде передана інформація, а також описує ситуацію в полі повідомлення.
Очікуваний результат:	Показана спливаюча підказка про збереження налаштувань.
Стан моделі після проведення випробувань:	Показана спливаюча підказка про збереження налаштувань.

Таблиця 5.10 – Натискання на кнопку SOS

Мета тесту	Перевірка функції «SOS виклик»
Початковий стан моделі	Відкрита головна сторінка застосування.
Вхідні дані:	
Схема проведення тесту:	Користувач натискає кнопку SOS
Очікуваний результат:	Виклик до служб і контактів, що задані в налаштуваннях кнопки SOS.
Стан моделі після проведення випробувань:	Виклик до служб і контактів, що задані в налаштуваннях кнопки SOS.

### Висновок до розділу

В даному розділі було наведено детальне керівництво користувача для ознайомлення перед роботою із застосуванням, у якому було показано приклади використання основних функцій.

Описано результати випробувань основних функцій та описана поведінка системи при введенні коректних і некоректних даних.

## ЗАГАЛЬНІ ВИСНОВКИ

У пояснювальній записці розглянуто детальний опис дипломного проекту, що присвячений розробці автоматизованої системи швидкого реагування на екстрені ситуації.

У розділі загальних положень розглядається основний процес діяльності користувача та функціональна модель за допомогою схеми варіантів використання. Крім того був наведений розгорнутий опис предметного середовища. Було розглянуто роботу оперативно-диспетчерської служби. Також було описано мету та призначення розробки.

У розділі наявних аналогів проаналізовано існуючі рішення, у порівнянні з якими виділені основні переваги та недоліки розробленого застосування.

У розділі інформаційного забезпечення описано, які вхідні дані поступають від користувача системою швидкого реагування на екстрені ситуації та які дані віддає система на виході. Була побудована схема структури бази даних, використана для збереження даних системи. Описано кожну таблицю та поля таблиць.

У математичному розділі наведено теоретичні відомості про задачу про найкоротший шлях, наведено детальний опис алгоритму Дейкстри та розглянуто приклад виконання алгоритму.

У розділі програмного та технічного забезпечення наведено опис використаних технологій, архітектури програмного забезпечення та побудована схема структурна класів. Були представлені основні компоненти програмного забезпечення у вигляді схеми структурної компонентів. Основна взаємодія класів програмного забезпечення були показані на схемі структурній послідовності. Представлені таблиці, що містять опис функцій, що представлені в схемі структурній класів.

У технологічному розділі наведено вичерпне керівництво користувача та екранні форми застосунку. Основні тестові сценарії та результат їх проходження описаний у розділі випробування програмного продукту.

Після проходження всіх етапів розробки та тестування можна зазначити, що програмний продукт відповідає всім функціональним вимогам, які до нього ставили у технічному завданні.

					ДП ІС-5208.1181-с.ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ПОСИЛАНЬ

1. Михайлов В. М. Організація взаємодії державних і муніципальних підрозділів екстреної допомоги населенню в системі 112 [Електронний ресурс] / В. М. Михайлов // «Державне управління: удосконалення та розвиток» : Електронне наукове фахове видання — Електронні дані — [Дніпро : Дніпропетров. держ. аграрно-економічний ун-т : ТОВ "ДКС Центр", 2015] — №4 — режим доступу: <http://www.dy.nayka.com.ua/?op=1&z=837>
2. Закон України «Про екстрену медичну допомогу» від 05.07.2012 р. № 5081-VI. Ст. 8 «Оперативно-диспетчерська служба центру екстреної медичної допомоги та медицини катастроф» // Голос України. – 2012. – № 152.
3. Організація диспетчерської служби екстреної медичної допомоги [Настанова] / [Мороз Є. Д., Близнюк М. Д., Гур'єв С. О. та ін.] — [Державний експертний центр Містерства охорони здоров'я України, Український науково-практичний центр екстреної медичної допомоги та медицини катастроф Міністерства охорони здоров'я України, Асоціація невідкладної медичної допомоги] — 2016 — 40с
4. Bankid.org.ua [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Українське бюро кредитних історій, 2005-2019. – Режим доступу: <https://bankid.org.ua/>
5. Прямая связь с полицией Дубая через приложение в смартфоне [Електронний ресурс] – Режим доступу: <https://dubai.in.ua/pryamaya-svyaz-s-policiej-dubaya-cherez-prilozhenie-v-smartfone.html>
6. Экстренные службы в Дубае будут приезжать на вызовы еще быстрее / [Електронний ресурс] – Режим доступу: <https://russianemirates.com/news/uae-news/ekstrennyye-sluzhby-v-dubaye-budut-priyezhat-na-vyzovy-yeshe-bystreje/>

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

7. Мобильное приложение для вызова экстренных служб появилось в Подмосковье / [Электронный ресурс] – Режим доступа: <https://news.softodrom.ru/ap/b28792.shtml>
8. Математика. Основы дискретной математики. Методические указания по изучению курса. / [А.Л. Пирозерский, Л.П. Пирозерская] – СПб.: Изд-во СПбГАСЭ, 2004. – 39 с.
9. Android Developers [Электронный ресурс] — Режим доступа: <https://developer.android.com/studio>
10. IntelliJ IDEA [Электронный ресурс] — Режим доступа: <https://www.jetbrains.com/idea/>
11. SQLite [Электронный ресурс] — Режим доступа: <https://www.sqlite.org/index.html>
12. Google Maps Api [Электронный ресурс] — Режим доступа: <https://developers.google.com/maps/documentation/>
13. Directions API [Электронный ресурс] — Режим доступа: <https://developers.google.com/maps/documentation/directions/start>
14. OpenStreetMaps [Электронный ресурс] — Режим доступа: <https://www.openstreetmap.org>
15. Overpass Turbo [Электронный ресурс] — Режим доступа: <https://overpass-turbo.eu>
16. Playground extract-osm-roads [Электронный ресурс] / Anvaka — Режим доступа: <https://github.com/anvaka/playground/tree/master/extract-osm-roads>
17. Gson [Электронный ресурс] — Режим доступа: <https://github.com/google/gson>
18. FirebaseUI [Электронный ресурс] — Режим доступа: <https://github.com/firebase/FirebaseUI-Android>
19. Firebase [Электронный ресурс] — Режим доступа: <https://firebase.google.com>





**CallFragment.java**

```
package info.androidhive.bottomnavigation.fragment;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.support.annotation.Nullable;
```

```
import android.support.v4.app.Fragment;
```

```
import android.support.v4.app.FragmentTransaction;
```

```
import android.view.LayoutInflater;
```

```
import android.view.View;
```

```
import android.view.ViewGroup;
```

```
import android.widget.Button;
```

```
import info.androidhive.bottomnavigation.MainActivity;
```

```
import info.androidhive.bottomnavigation.R;
```

```
public class CallFragment extends Fragment {
```

```
    private Button btn101;
```

```
    private Button btn102;
```

```
    private Button btn103;
```

```
    private Button btn104;
```

```
    private Button btn101call;
```

```
    private Button btn102call;
```

```
    private Button btn103call;
```

```
    private Button btn104call;
```

```
    private SosButtonFragment sosButtonFragment;
```

```
    private CallProcessFragment callProcessFragment;
```

					ДП ІС-5208.1181-с.ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

```
private boolean isSosButtonDisplaying;
CallSharedViewModel sharedViewModel;
```

```
public CallFragment() {
    // Required empty public constructor
}
```

```
public static CallFragment newInstance(String param1, String param2) {
    CallFragment fragment = new CallFragment();
    Bundle args = new Bundle();
    fragment.setArguments(args);
    return fragment;
}
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}
```

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View rootView = inflater.inflate(R.layout.fragment_call, container, false);

    PackageManager packageManager = getPackageManager();
    boolean telephonySupported =
packageManager.hasSystemFeature(PackageManager.FEATURE_TELEPHONY);
```

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

```
boolean gsmSupported =
packageManager.hasSystemFeature(PackageManager.FEATURE_TELEPHONY_
GSM);
```

```
boolean cdmaSupported =
packageManager.hasSystemFeature(PackageManager.FEATURE_TELEPHONY_
CDMA);
```

```
boolean canCall = telephonySupported && gsmSupported &&
cdmaSupported;
```

```
Button btn101 = rootView.findViewById(R.id.button101);
```

```
Button btn102 = rootView.findViewById(R.id.button102);
```

```
Button btn103 = rootView.findViewById(R.id.button103);
```

```
Button btn104 = rootView.findViewById(R.id.button104);
```

```
Button btn101call = rootView.findViewById(R.id.button101call);
```

```
Button btn102call = rootView.findViewById(R.id.button102call);
```

```
Button btn103call = rootView.findViewById(R.id.button103call);
```

```
Button btn104call = rootView.findViewById(R.id.button104call);
```

```
sharedViewModel =
```

```
ViewModelProviders.of(getActivity()).get(CallSharedViewModel.class);
```

```
sosButtonFragment = new SosButtonFragment();
```

```
callProcessFragment = new CallProcessFragment();
```

```
insertSosButtonFragment();
```

```
btn101.setOnClickListener(new View.OnClickListener() {
```

```
@Override
```

```

    public void onClick(View view) {
        sharedViewModel.select(101);
        changeFragment();
    }
});

btn102.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sharedViewModel.select(102);
        changeFragment();
    }
});

btn103.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sharedViewModel.select(103);
        changeFragment();
    }
});

btn104.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sharedViewModel.select(104);
        changeFragment();
    }
});

```

```
if (canCall){
```

```
    btn101call.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent call = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:101"));
            startActivity(call);
        }
    });
```

```
    btn102call.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent call = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:102"));
            startActivity(call);
        }
    });
```

```
    btn103call.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent call = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:103"));
            startActivity(call);
        }
    });
```

```
    btn104call.setOnClickListener(new View.OnClickListener() {
        @Override
```

					ДП ІС-5208.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

```

        public void onClick(View view) {
            Intent call = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:104"));
            startActivity(call);
        }
    });
}

```

```

    return rootView;
}

```

```

private void changeFragment()
{
    if (isSosButtonDisplaying) {
        displayCallProcessFragment();
    } else {
        displaySosButtonFragment();
    }
    isSosButtonDisplaying = !isSosButtonDisplaying;
}

```

```

protected void displaySosButtonFragment() {
    FragmentTransaction fragmentTransaction =
getChildFragmentManager().beginTransaction();
    if (sosButtonFragment.isAdded()) {
        fragmentTransaction.show(sosButtonFragment);
    } else {
        fragmentTransaction.add(R.id.call_child_fragment_container,
sosButtonFragment, "sosButtonFragment");
    }
}

```

```

    }

```

```

    if (callProcessFragment.isAdded()) {
        fragmentTransaction.hide(callProcessFragment);
    }
    fragmentTransaction.commit();
}

```

```

protected void displayCallProcessFragment() {
    FragmentTransaction fragmentTransaction =
getChildFragmentManager().beginTransaction();
    if (callProcessFragment.isAdded()) {
        fragmentTransaction.show(callProcessFragment);
    } else {
        fragmentTransaction.add(R.id.call_child_fragment_container,
callProcessFragment, "callProcessFragment");
    }
}

```

```

    if (sosButtonFragment.isAdded()) {
        fragmentTransaction.hide(sosButtonFragment);
    }
    fragmentTransaction.commit();
}

```

```

@Override

```

```

public void onViewCreated(View view, @Nullable Bundle savedInstanceState)
{
    super.onViewCreated(view, savedInstanceState);
}

```



```

        insertSosButtonFragment();
        isSosButtonDisplaying = true;
    }

```

```

    private void insertCallProcessFragment() {
        FragmentTransaction transaction =
        getChildFragmentManager().beginTransaction();
        transaction.replace(R.id.call_child_fragment_container,
        callProcessFragment).commit();
    }

```

```

    private void insertSosButtonFragment() {
        FragmentTransaction transaction =
        getChildFragmentManager().beginTransaction();
        transaction.replace(R.id.call_child_fragment_container, sosButtonFragment);
    }
}

```

### CallSharedViewModel.java

```

import android.arch.lifecycle.LiveData;
import android.arch.lifecycle.MutableLiveData;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import info.androidhive.bottomnavigation.R;

```

```

public class CallSharedViewModel extends ViewModel {

```

```
private final MutableLiveData<Integer> selected = new  
MutableLiveData<Integer>();  
  
public void select(int emergency) {  
    selected.setValue(item);  
}  
  
public LiveData<Integer> getSelected() {  
    return selected;  
}  
  
}
```

					ДП ІС-5208.1181-с.ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

## Додаток А

**Тексти програмного коду***Автоматизована система швидкого реагування на екстрені ситуації**для мобільної платформи*

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*9 арк, 68 Кб*

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019 року

Змн.	Арк.	№ докум.	Підпис	Дата



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_ М.О. Солдатова  
(підпис) (ініціали, прізвище)

“16” квітня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ О.А. Павлов  
(підпис) (ініціали, прізвище)

“17” квітня 2019 р.

Автоматизована система швидкого реагування на екстрені ситуації  
для мобільної платформи

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр ДП ІС-5208.1181-с.ТЗ

на 11 сторінках

Київ – 2019 року

## ЗМІСТ

1.1	Повне найменування системи та її умовне позначення .....	3
1.2	Найменування організації-замовника та організації-учасника робіт.....	3
1.3	Перелік документів, на підставі яких створюється система .....	3
1.4	Планові терміни початку і закінчення роботи зі створення системи.....	4
2	ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ СИСТЕМИ .....	5
2.1	Призначення системи .....	5
2.2	Цілі створення системи .....	5
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ .....	6
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	7
4.1	Вимоги до функціональних характеристик .....	7
4.2	Вимоги до надійності .....	8
4.3	Вимоги до складу і параметрів технічних засобів.....	9
5	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	10
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ .....	11
6.1	Види випробувань.....	10

					<b>ДП ІС-5208.1181-с.ТЗ</b>			
Зм.	Арк.	Прізвище	Підпис	Дата	<div>Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи</div>			
Розроб.		Дорогін І.Р.						
Перевірив.		Солдатова М.О.						
Н. кон.		Халус О.А.						
Затв.		Павлов О.А.			<div>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</div>			
					Літ.	Лист	Листів	
						2	11	

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Повне найменування системи та її умовне позначення

Повна назва системи: Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи

### 1.2 Найменування організації-замовника та організації-учасника робіт

Замовником проекту є кафедра Автоматизованих систем обробки інформації та управління НТУУ «КПІ ім. Ігоря Сікорського». Представником замовника є Солдатова Марія Олександрівна.

Розробником системи є студент факультету інформатики та обчислювальної техніки НТУУ «КПІ ім. Ігоря Сікорського»: студент групи ІС-52 Дорогін Ілля Ростиславович.

### 1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

					ДП ІС-5208.1181-с.ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над інформаційною системою підтримки навчальної діяльності студента – 5 лютого 2019 рік.

Плановий термін по закінченню роботи над автоматизованою системою швидкого реагування на екстрені ситуації для мобільної платформи – не пізніше 1 червня 2019 року.

					ДП ІС-5208.1181-с.ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		



## 2 ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ СИСТЕМИ

### 2.1 Призначення системи

Призначенням системи є забезпечення швидкого та інформативного способу для виклику екстрених служб через мобільний застосунок.

### 2.2 Цілі створення системи

Цілями розробки системи є:

- прискорення процесу виклику екстрених служб;
- збільшення інформативності даних про виклик;
- автоматизація процесу диспетчеризації екстрених ситуацій.

Для досягнення поставлених цілей необхідно реалізувати наступні задачі:

- виклик екстрених служб
- пошук найближчого закладу потрібного типу та побудова найкоротшого маршруту для бригади екстреного реагування
- перегляд маршруту бригади на карті
- чат для комунікації з диспетчером;
- перегляд довідкової інформації;
- можливість створення власної медичної картки;
- кнопка SOS з можливістю її настройки.

					ДП ІС-5208.1181-с.ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Працювати з застосуванням можуть всі користувачі зі смартфоном на базі ОС Android версії 4.0 (Ice Cream Sandwich) або вищої. З його допомогою можна викликати екстренні служби на актуальне місце подій, яке визначатиметься автоматично службами геолокації, створити медичну картку (задати особисті медичні дані, які можуть бути потрібними екстреним службам для надання правильної допомоги (наприклад, наявність алергічних реакцій на ті чи інші речовини), а також отримати інформацію про надання першої допомоги та дії при різних надзвичайних ситуаціях.

Об'єктом автоматизації є процес диспетчеризації екстрених ситуацій.

					ДП ІС-5208.1181-с.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Застосування має надати зручний та швидкий спосіб виклику екстрених служб. Застосування має виконувати наступні функції:

- Система повинна надати користувачу можливість реєстрації та входу.
- Система повинна надати користувачу можливість викликати екстрену службу.
  - 1) Система повинна надати користувачу можливість обрати потрібну екстрену службу.
  - 2) Система повинна надати користувачу можливість обрати тип ситуації.
  - 3) Система повинна надати користувачу можливість надати інформацію про ситуацію.
    - a) Система повинна надати користувачу можливість ввести текст.
    - b) Система повинна надати користувачу можливість прикріпити фото.
    - c) Система повинна надати користувачу можливість додавати постраждалих.
  - 4) Після виклику система повинна представити користувачу довідкову інформацію про надання першої допомоги та дії при даній ситуації ситуації.
  - 5) Система повинна надати користувачу можливість комунікації (чат) з диспетчером.
- Система повинна надати користувачу можливість переглядати довідкову інформацію про надання першої допомоги та дії при різних надзвичайних ситуаціях.

– Система повинна надати користувачу можливість створення власної медичної картки.

1) Система повинна надати користувачу заповнити такі поля медичної картки: хвороби, алергічні реакції, медикаменти, група крові, вага, зріст.

– Система повинна надати користувачу можливість швидко викликати допомогу кнопкою SOS.

1) Система повинна надати можливість налаштувати кнопку SOS.

a) Система повинна надати користувачу можливість обрати екстрені служби, до яких буде надісланий запит.

b) Система повинна надати користувачу можливість обрати контакти, яким буде відправлено повідомлення.

c) Система повинна надати користувачу можливість ввести повідомлення, яке буде надіслане.

## 4.2 Вимоги до надійності

Програма повинна зберігати працездатність і забезпечувати відновлення своїх функцій при виникненні наступних позаштатних ситуацій:

– при помилках в роботі апаратних засобів.

Програмний продукт повинен поєднувати надійність та функціональність. У разі виникнення аварійних ситуацій необхідно сповіщати користувача та надавати інструкцію для подальших дій. Будь-які аварійні ситуації мають бути задокументовані у звіті, який при необхідності надсилається розробнику для визначення причини збою в роботі та усуненні помилок, які могли привести до нестабільної роботи програмного продукту.

					ДП ІС-5208.1181-с.ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

### 4.3 Вимоги до складу і параметрів технічних засобів

Склад, структура і способи організації даних в системі повинні бути визначені на етапі технічного проектування.

Структура технічних засобів визначається виходячи із можливості їх забезпечити виконання встановлених операцій процесу технічного обслуговування.

Для забезпечення правильної та стабільної роботи даного програмного продукту висуваються такі вимоги:

- смартфон на базі ОС Android;
- версія Android 4.0(Ice Cream Sandwich) або вища.

					ДП ІС-5208.1181-с.ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з автоматизації системи швидкого реагування на екстрені ситуації.

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	07.02.2019	
2.	Розробка сценарію роботи	12.02.2019	
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	20.02.2019	
4.	Узгодження з керівником інтерфейсу користувача	02.03.2019	
5.	Розробка інформаційного забезпечення	10.03.2019	
6.	Розробка програмного забезпечення	10.04.2019	
7.	Налагодження програми	20.04.2019	
8.	Тестування програми	16.05.2019	
9.	Здача готового програмного продукту замовнику	30.05.2019	

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

### 6.1 Види випробувань

Види, склад, об'єм і методи випробувань програмного продукту повинні бути викладені в програмі і методиці випробувань системи, що розробляється в складі робочої документації.

Випробування мають проводитись на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

					ДП ІС-5208.1181-с.ТЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_  
(підпис) М.О. Солдатова  
(ініціали, прізвище)

“13” травня 2019 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_  
(підпис) О.А.Павлов  
(ініціали, прізвище)

“14” травня 2019 р.

Автоматизована система швидкого реагування на екстрені ситуації  
для мобільної платформи

**ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ**

Шифр ДП ІС-5208.1181-с.ПМВ

на 15 сторінках

Київ – 2019 року



## ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАННЯ .....	3
1.1	Найменування програми.....	3
1.2	Область застосування .....	3
1.3	Умовне позначення програми .....	3
2	МЕТА ВИПРОБУВАНЬ .....	4
3	ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ .....	5
3.1	Вимоги до функціональних характеристик .....	5
3.1.1	Вимоги до складу виконуваних функцій.....	5
4	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ .....	6
5	СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ .....	7
6	МЕТОДИ ВИПРОБУВАНЬ.....	8

					ДП ІС-5208.1181-с.ПМВ											
Зм.	Арк.	Прізвище	Підпис	Дата	Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи					Літ.		Лист		Листів		
Розроб.		Дорогін І.Р.											2		10	
										КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52						
Перевірив.		Солдатова М.О.														
Н. кон.		Халус О.А.														
Затв.		Павлов О.А.														

## 1 ОБ'ЄКТ ВИПРОБУВАННЯ

### 1.1 Найменування програми

Повне найменування програми: Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи.

### 1.2 Область застосування

Застосування призначене для виклику екстрених служб на місце пригоди за допомогою Android-смартфону.

### 1.3 Умовне позначення програми

Умовне позначення: «EmergencyApp».

					ДП ІС-5208.1181-с.ПМВ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 МЕТА ВИПРОБУВАНЬ

Метою випробувань є перевірка відповідності функцій автоматизованої системи швидкого реагування на екстрені ситуації вимогам технічного завдання.

					ДП ІС-5208.1181-с.ПМВ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Вимоги до функціональних характеристик

Функціональні вимоги:

- Система повинна надати користувачу можливість реєстрації та входу.
- Система повинна надати користувачу можливість викликати екстрену службу.
  - 1) Система повинна надати користувачу можливість обрати потрібну екстрену службу.
  - 2) Система повинна надати користувачу можливість обрати тип ситуації.
  - 3) Система повинна надати користувачу можливість надати інформацію про ситуацію.
    - а) Система повинна надати користувачу можливість ввести текст.
    - б) Система повинна надати користувачу можливість прикріпити фото.
    - с) Система повинна надати користувачу можливість додавати постраждалих.
  - 4) Після виклику система повинна представити користувачу довідкову інформацію про надання першої допомоги та дії при даній ситуації ситуації.
  - 5) Система повинна надати користувачу можливість комунікації (чат) з диспетчером.
- Система повинна надати користувачу можливість переглядати довідкову інформацію про надання першої допомоги та дії при різних надзвичайних ситуаціях.

– Система повинна надати користувачу можливість створення власної медичної картки.

1) Система повинна надати користувачу заповнити такі поля медичної картки: хвороби, алергічні реакції, медикаменти, група крові, вага, зріст.

– Система повинна надати користувачу можливість швидко викликати допомогу кнопкою SOS.

1) Система повинна надати можливість налаштувати кнопку SOS.

a) Система повинна надати користувачу можливість обрати екстрені служби, до яких буде надісланий запит.

b) Система повинна надати користувачу можливість обрати контакти, яким буде відправлено повідомлення.

c) Система повинна надати користувачу можливість ввести повідомлення, яке буде надіслане.

### 3.1.1 Вимоги до складу виконуваних функцій

Функціонал повинен забезпечувати можливість виконання перерахованих нижче функцій:

- виклик екстреної служби обраного типу
- перегляд результатів виклику
- перегляд довідкових даних
- налаштування кнопки SOS
- створення особистої медичної картки

#### 4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація має складатися з керівництва користувача та вихідних текстів програмного коду.

					ДП ІС-5208.1181-с.ПМВ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

Етапи випробувань:

- ознайомчий;
- виконавчий.

На ознайомчому етапі проводиться:

- перевірка комплектності програмної документації;
- перевірка комплектності складу технічних і програмних засобів.

Під час виконавчого етапу проводиться:

- перевірка відповідності технічних характеристик системи;
- перевірка ступеню виконання вимог функціонального призначення системи.

Функції, що підлягають перевірці:

- авторизація/реєстрація
- виклик екстреної служби обраного типу
- перегляд результатів виклику
- перегляд довідкових даних
- налаштування кнопки SOS
- створення особистої медичної картки

## 6 МЕТОДИ ВИПРОБУВАНЬ

Після проведення тестування було перевірено основну функціональність системи. В представлених таблицях наведено результати випробувань основних функцій програмного продукту.

Таблиця 6.1 – Реєстрація нового користувача

Мета тесту	Перевірка функції «Реєстрація»
Початковий стан моделі	Відкрита сторінка реєстрації.
Вхідні дані:	Прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль.
Схема проведення тесту:	Заповнити наступні поля: прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль. Натиснути на кнопку «Реєстрація».
Очікуваний результат:	Реєстрація пройдена успішно, відкрита сторінка авторизації..
Стан моделі після проведення випробувань:	Реєстрація пройдена успішно, відкрита сторінка авторизації..



Таблиця 6.2 – Перевірка заповнення всіх полів при реєстрації

<b>Мета тесту</b>	<b>Перевірка функції «Перевірка заповнення усіх полів при реєстрації»</b>
Початковий стан моделі	Відкрита сторінка реєстрації
Вхідні дані:	Прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль за винятком деяких
Схема проведення тесту:	Заповнити наступні поля, залишивши деякі з них незаповненими: прізвище, ім'я, по-батькові, дата народження, місто, місце проживання, номер телефону, електронна пошта, пароль.
Очікуваний результат:	Незаповнені поля підсвічуються червоним кольором
Стан моделі після проведення випробувань:	Незаповнені поля підсвічуються червоним кольором

Таблиця 6.3 – Вхід користувача у систему

<b>Мета тесту</b>	<b>Перевірка функції «Авторизація»</b>
Початковий стан моделі	Відкрита сторінка авторизації
Вхідні дані:	Електронна пошта та пароль користувача
Схема проведення тесту:	Заповнити поля електронної пошти та паролю. Натиснути на кнопку «Вхід».
Очікуваний результат:	Відкрита сторінка виклику екстрених служб.

## Продовження таблиці 6.3

<b>Мета тесту</b>	<b>Перевірка функції «Авторизація»</b>
Стан моделі після проведення випробувань:	Відкрита сторінка виклику екстрених служб.

Таблиця 6.4 – Перевірка правильності введених даних

<b>Мета тесту</b>	<b>Перевірка функції «Перевірка правильності введених даних»</b>
Початковий стан моделі	Відкрита сторінка авторизації
Вхідні дані:	Електронна пошта та пароль користувача
Схема проведення тесту:	При заповненні полів електронної пошти та паролю, ввести неправильні дані. Натиснути на кнопку «Вхід».
Очікуваний результат:	Повідомлення «Введено некоректні дані».
Стан моделі після проведення випробувань:	Повідомлення «Введено некоректні дані».

Таблиця 6.5 – Вибір екстренної служби

<b>Мета тесту</b>	<b>Перевірка функції «Вибір екстренної служби»</b>
Початковий стан моделі	Відкрита головна сторінка застосування.
Вхідні дані:	

## Продовження таблиці 6.5

<b>Мета тесту</b>	<b>Перевірка функції «Вибір екстренної служби»</b>
Схема проведення тесту:	Натиснути на кнопку однієї з екстрених служб
Очікуваний результат:	Відрита сторінка виклику обраної екстреної служби.
Стан моделі після проведення випробувань:	Відрита сторінка виклику обраної екстреної служби.

## Таблиця 6.6 – Виклик екстренної служби

<b>Мета тесту</b>	<b>Перевірка функції «Виклик екстренної служби»</b>
Початковий стан моделі	Відкрита сторінка виклику екстренної служби
Вхідні дані:	Тип ситуації, додаткова інформація, постраждалі.
Схема проведення тесту:	Користувач з випадного списку обирає потрібну ситуацію, заповнює поле «додаткова інформація» та додає користувачів до списку постраждалих.
Очікуваний результат:	Відкрита сторінка очікування з довідковою інформацією по обраному типу ситуації та чатом з диспетчером.

## Продовження таблиці 6.6

<b>Мета тесту</b>	<b>Перевірка функції «Виклик екстренної служби»</b>
Стан моделі після проведення випробувань:	Відкрита сторінка очікування з довідковою інформацією по обраному типу ситуації та чатом з диспетчером.

## Таблиця 6.7 – Перегляд маршруту бригади

<b>Мета тесту</b>	<b>Перевірка функції «Перегляд маршрути бригади»</b>
Початковий стан моделі	Відкрита сторінка очікування
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Маршрут бригади».
Очікуваний результат:	Відкрита сторінка з мапою, на якій відображено маршрут бригади до місця виклику
Стан моделі після проведення випробувань:	Відкрита сторінка з мапою, на якій відображено маршрут бригади до місця виклику

Таблиця 6.8 – Створення медичної картки

<b>Мета тесту</b>	<b>Перевірка функції «Створення медичної картки»</b>
Початковий стан моделі	Відкрита сторінка медичної картки
Вхідні дані:	Хвороби, алергічні реакції, медикаменти, група крові, вага, зріст.
Схема проведення тесту:	Користувач заповнює доступні поля поля
Очікуваний результат:	Показана спливаюча підказка про збереження медичної картки. Оновлена медична картка.
Стан моделі після проведення випробувань:	Показана спливаюча підказка про збереження медичної картки. Оновлена медична картка.

Таблиця 6.9 – Налаштування кнопки SOS

<b>Мета тесту</b>	<b>Перевірка функції «Налаштування кнопки SOS»</b>
Початковий стан моделі	Відкрита сторінка налаштувань кнопки SOS
Вхідні дані:	Екстренні служби, екстрені контакти, примітка
Схема проведення тесту:	Користувач обирає потрібні екстренні служби та контакти, до яких буде передана інформація, а також описує ситуацію в полі повідомлення.

## Продовження таблиці 6.9

<b>Мета тесту</b>	<b>Перевірка функції «Налаштування кнопки SOS»</b>
Очікуваний результат:	Показана спливаюча підказка про збереження налаштувань.
Стан моделі після проведення випробувань:	Показана спливаюча підказка про збереження налаштувань.

## Таблиця 6.10 – Натискання на кнопку SOS

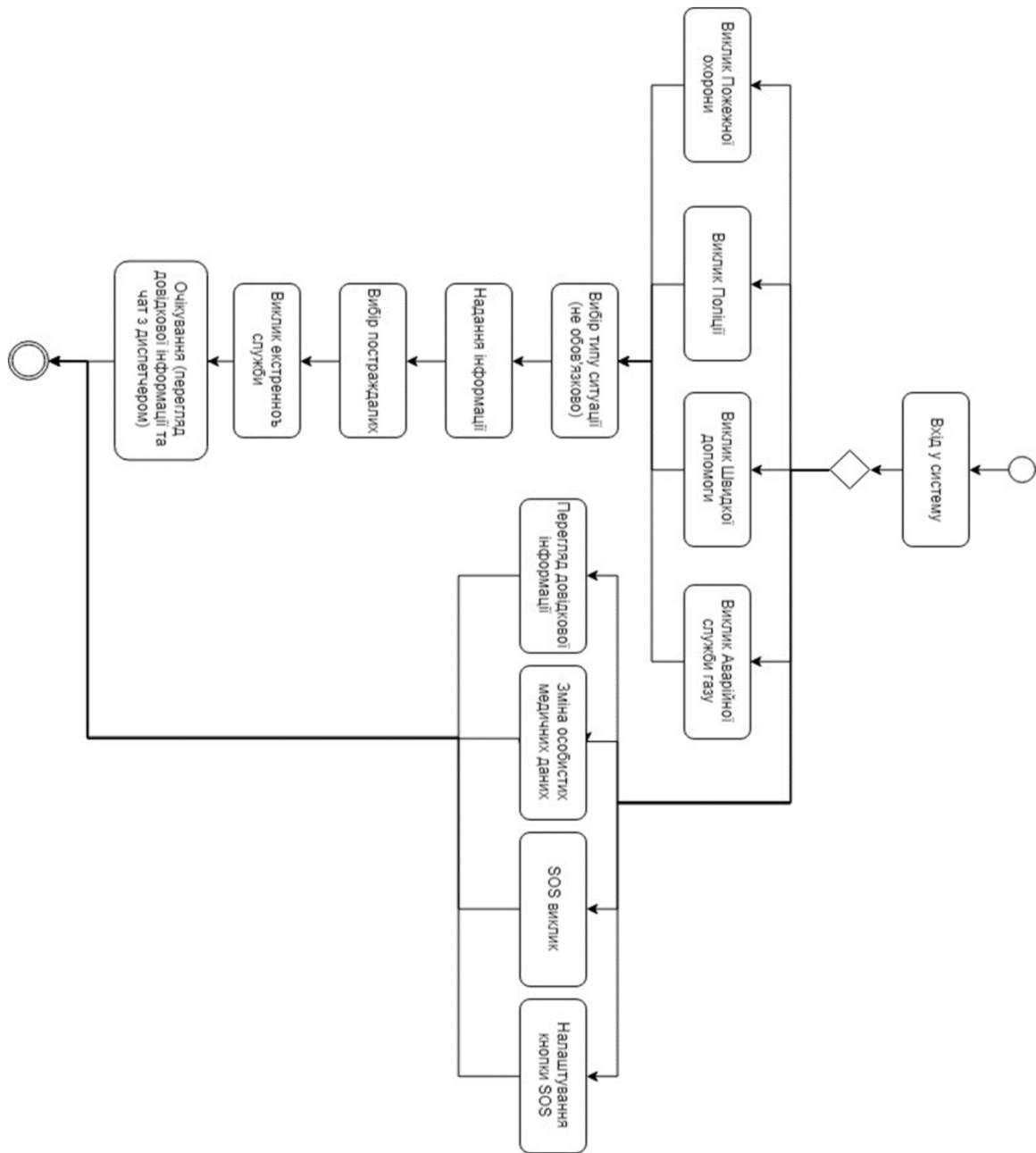
<b>Мета тесту</b>	<b>Перевірка функції «SOS виклик»</b>
Початковий стан моделі	Відкрита головна сторінка застосування.
Вхідні дані:	
Схема проведення тесту:	Користувач натискає кнопку SOS
Очікуваний результат:	Виклик до служб і контактів, що задані в налаштуваннях кнопки SOS.
Стан моделі після проведення випробувань:	Виклик до служб і контактів, що задані в налаштуваннях кнопки SOS.

# **Графічний матеріал до дипломного проекту**

на тему: Автоматизована система швидкого реагування на екстрені ситуації  
для мобільної платформи

---

Київ – 2019 року



ДП ІС-5208.1181-с.ССД

Схема структурна діяльності

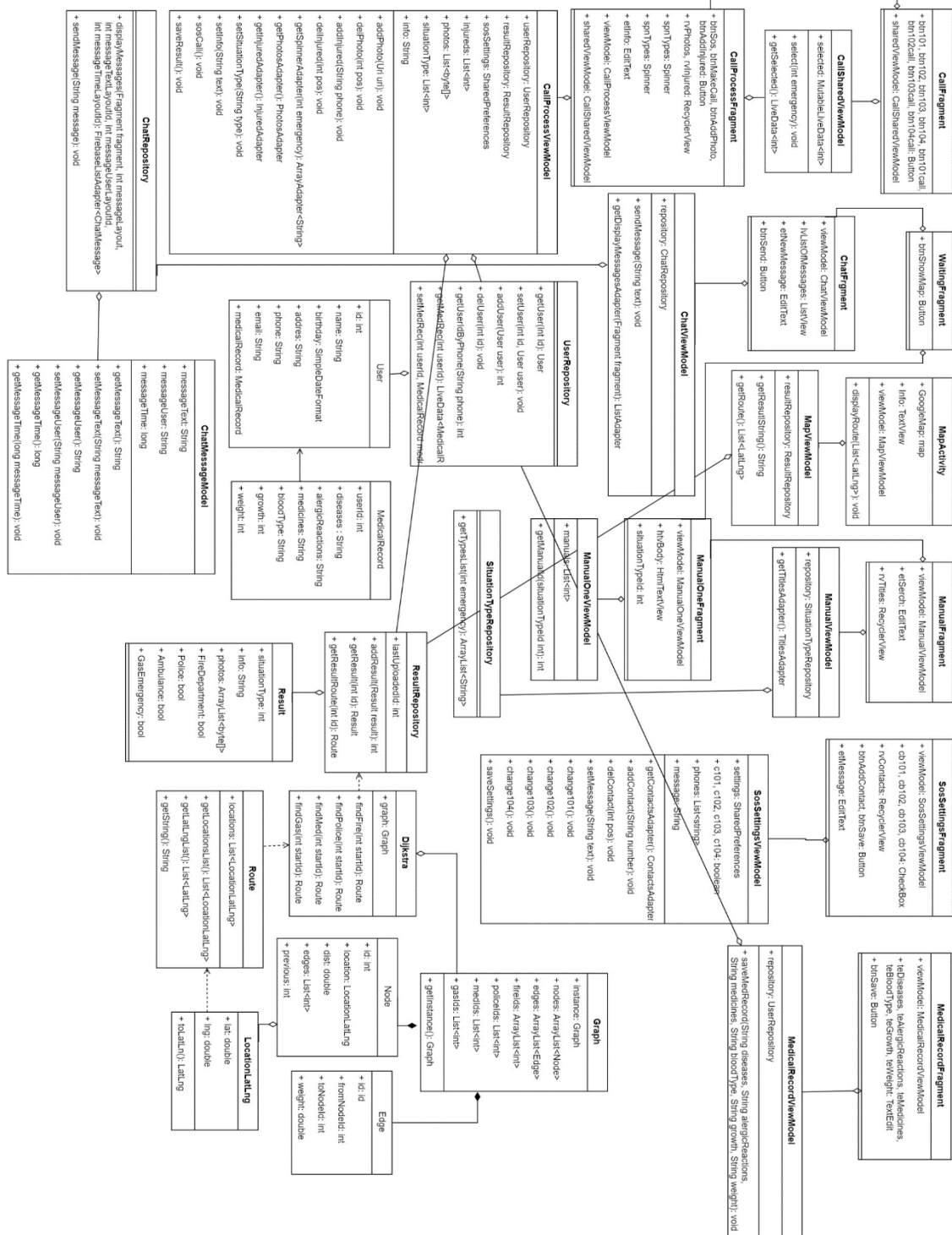
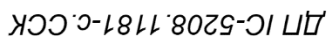
Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи

Літера	Маса	Масштаб
Аркуш 1		Аркушів 1

КПІ ім. Ігоря Сікорського  
кафедра АСОІУ гр. ІС-52

Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Дорогін І.Р.		
Перевірив		Солдатова М.О.		
Т. кон.				
Н. кон.		Халус О.А.		
Затвердив		Солдатова М.О.		





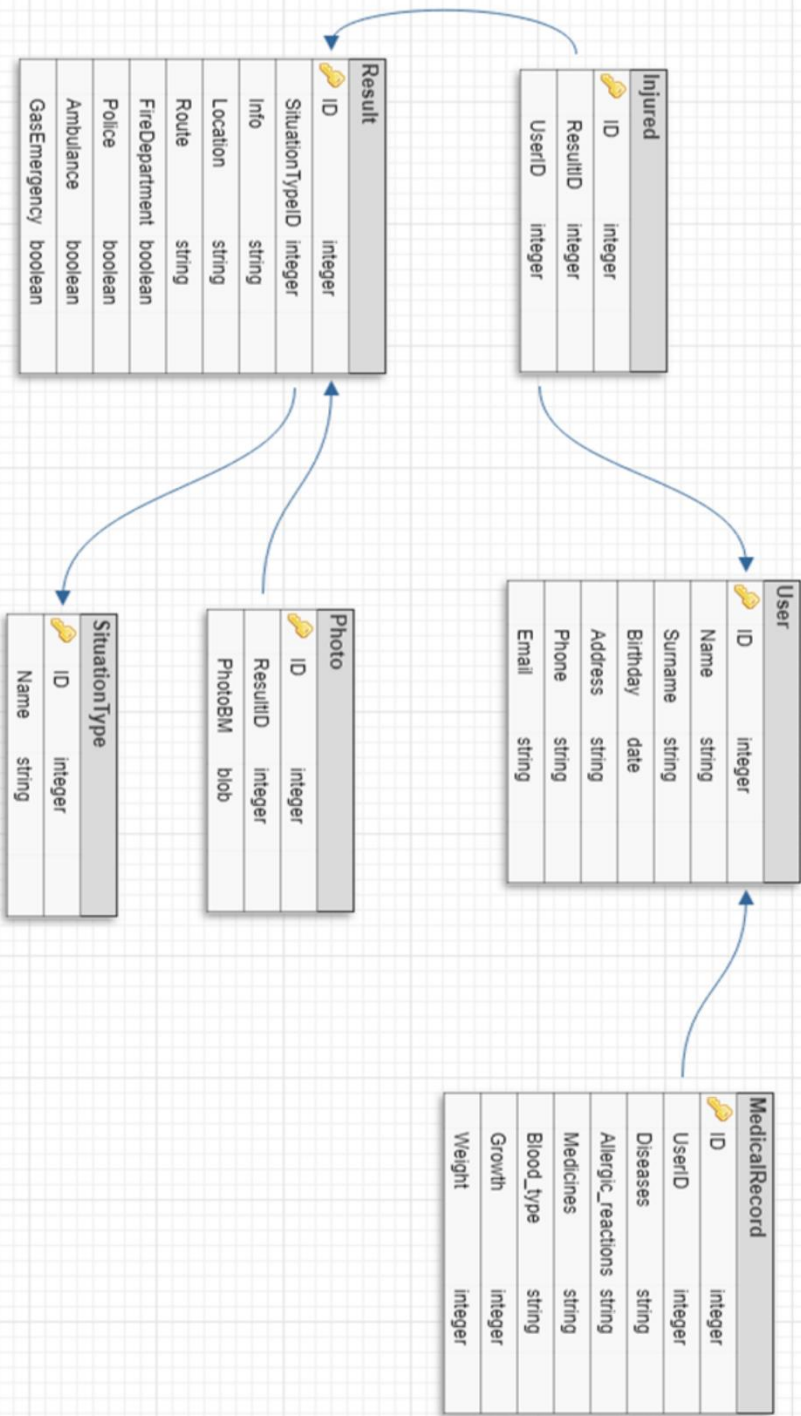
ДП IC-5208.1181-с.ССК

Схема стректурна класів  
програмного забезпечення

Автоматизована система швидкого  
реагування на екстрені ситуації для  
мобільної платформи

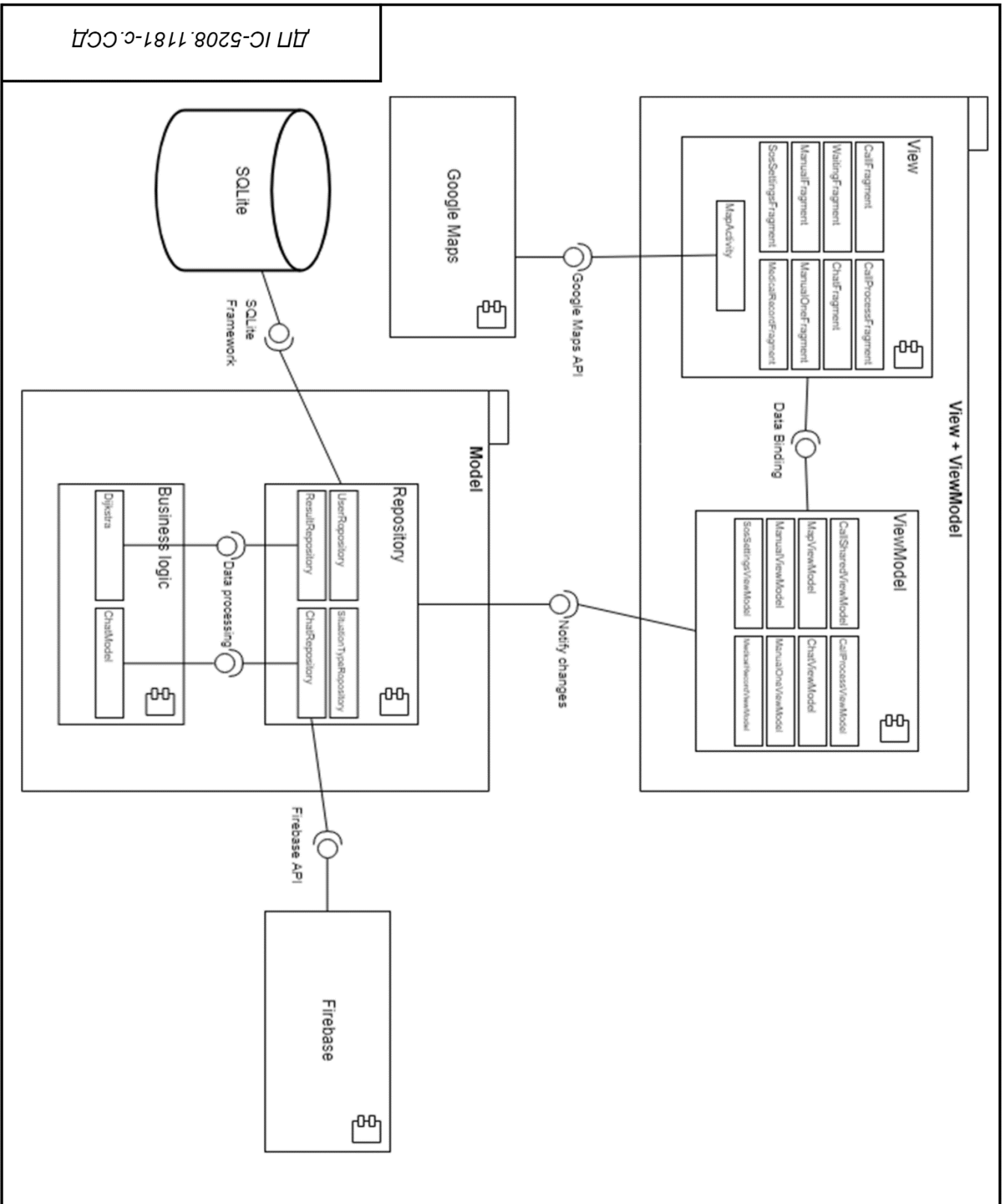
Літера			Маса		Масштаб	
Аркуш 1			Аркушів 1			
<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52</i>						

Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Дорогін І.Р.		
Перевірив		Солдатова М.О.		
Т. кон.				
Н. кон.		Халус О.А.		
Затвердив		Солдатова М.О.		



ДП IC-5208.1181-с.СБД							
Схема бази даних					Літера	Маса	Масштаб
					Аркуш 1	Аркушів 1	
Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. IC-52		

Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Дорогін І.Р.		
Перевірив		Солдатова М.О.		
Т. кон.				
Н. кон.		Халус О.А.		
Затвердив		Солдатова М.О.		



ДП ІС-5208.1181-с.ССК

ДП ІС-5208.1181-с.ССК

					Літера			Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Дорогін І.Р.							
Перевірив		Солдатова М.О.							
Т. кон.									
Н. кон.		Халус О.А.							
Затвердив		Солдатова М.О.							

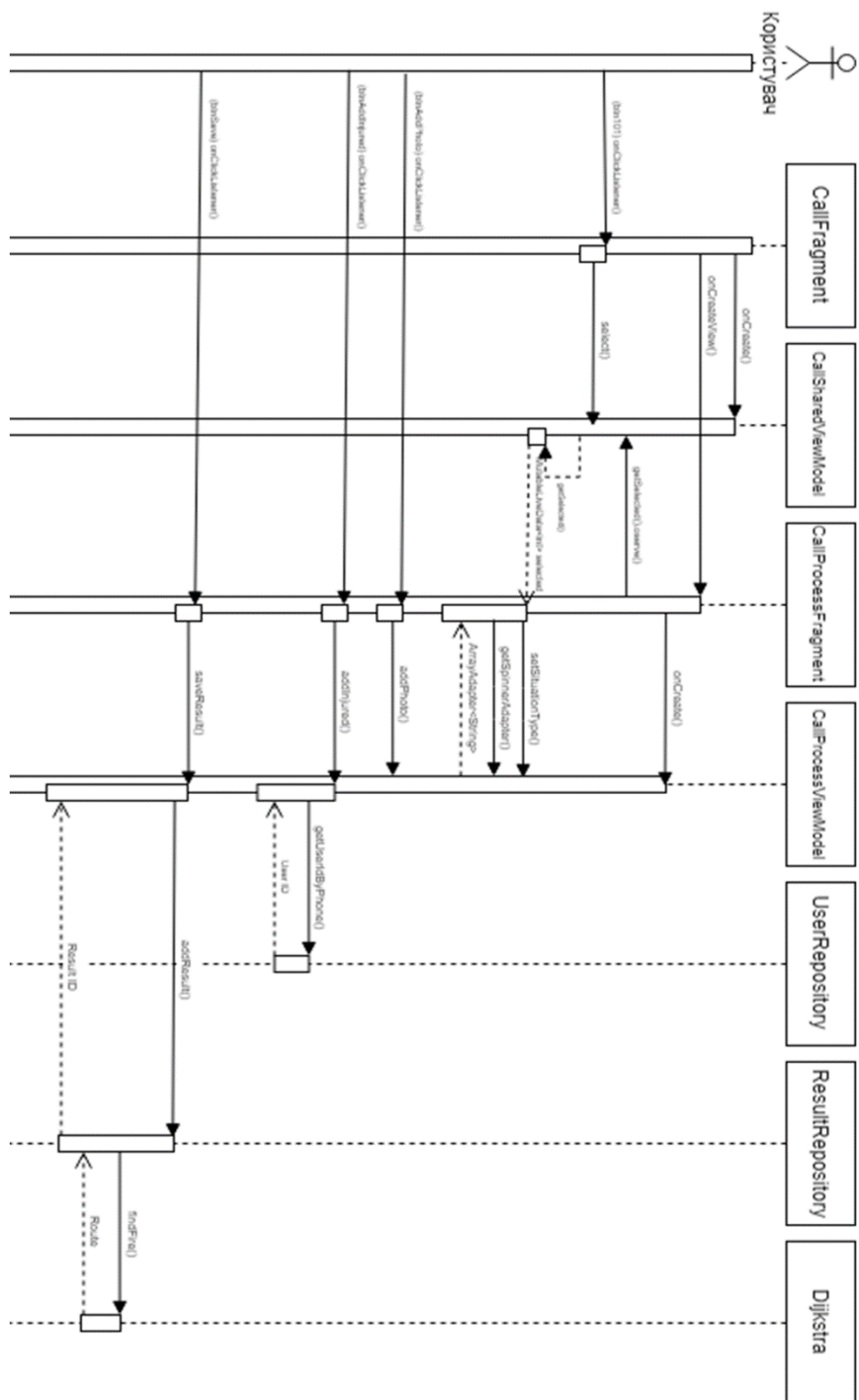
Схема стректурна компонентів програмного забезпечення

Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи

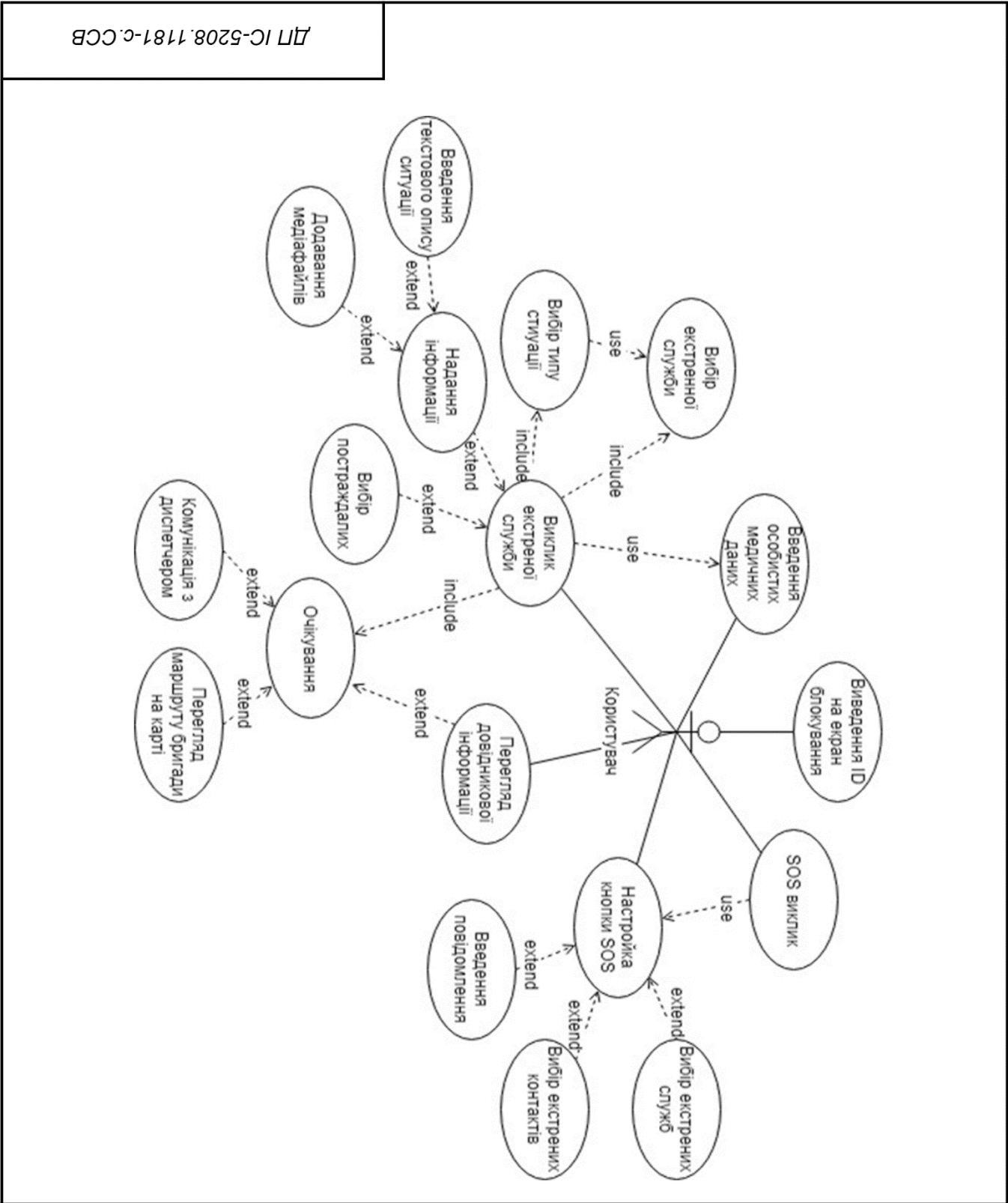
Аркуш 1

Аркушів 1

КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52



					ДП ІС-5208.1181-с.ССП					
Зм.	Арк.	№ документа	Підпис	Дата	Схема стректурна поїлдовності	Літера			Маса	Масштаб
Розробив	Дорогін І.Р.									
Перевірив	Солдатов М.О.									
Т. кон.					Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи	Аркуш 1			Аркушів 1	
Н. кон.	Халус О.А.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52				
Затвердив	Солдатов М.О									



					ДП ІС-5208.1181-с.ССВ						
					Схема варіантів використань	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Дорогін І.Р.									
Перевірив		Солдатова М.О.									
Т. кон.					Автоматизована система швидкого реагування на екстрені ситуації для мобільної платформи	Аркуш 1		Аркушів 1			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52
Н. кон.		Халус О.А.									
Затвердив		Солдатова М.О									